InterPlanetary File System pada *Digital Evidence Cabinet* berbasis *Hyperledger Fabric* untuk Manajemen Bukti Digital

Jefrul Hanafi
18917114

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Forensika Digital
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2022

Lembar Pengesahan Pembimbing

***InterPlanetary File System* pada *Digital Evidence Cabinet* berbasis *Hyperledger Fabric* untuk Manajemen Bukti Digital**

Jefrul Hanafi
18917114

Yogyakarta, Mei 2022

Pembimbing I	Pembimbing II
Dr. Yudi Prayudi, S.Si., M.Kom.	Ahmad Luthfi, S.Kom., M.Kom., Ph.D.

Lembar Pengesahan Penguji

***InterPlanetary File System* pada *Digital Evidence Cabinet* berbasis *Hyperledger Fabric* untuk Manajemen Bukti Digital**

Jefrul Hanafi

18917114

Yogyakarta, Mei 2022

Tim Penguji,

Dr. Yudi Prayudi, S.Si., M.Kom.
Ketua

Ahmad Luthfi, S.Kom., M.Kom., Ph.D.
Anggota I

Dr. Ir. Bambang Sugiantoro, S.Si., M.T.
Anggota II

Mengetahui,

Ketua Program Studi Teknik Informatika Program Magister

Universitas Islam Indonesia

Izzati Muhimmah, S.T., M.Sc., Ph.D.

Abstrak

InterPlanetary File System pada *Digital Evidence Cabinet* berbasis *Hyperledger Fabric* untuk Manajemen Bukti Digital

Dematerialisasi bukti fisik adalah aset digitalisasi informasi, dimana informasi dari bukti fisik menjadi lebih penting dari bukti fisik itu sendiri. Pada prinsipnya, untuk melestarikan bukti digital, diperlukan sistem penyimpanan terdistribusi yang akurat dan andal yang dikemas dengan istilah digitalisasi aset yang terangkum dalam dokumen *chain of custody* (CoC). Selain menangani segala kebutuhan yang berkaitan dengan sistem penyimpanan, satu hal yang tak kalah pentingnya adalah fokus pada kemudahan pendistribusian bukti di jaringan secara aman dan terpercaya. Ini adalah salah satu bagian terpenting dalam menafsirkan efektivitas investigasi forensik digital. Namun, beberapa permasalahan muncul terkait konsep pengelolaan penyimpanan aset bukti digital yang masih belum dapat didistribusikan, dan sulit dilacak. Tujuan dari penelitian ini adalah untuk melengkapi konsep *Digital Evidence Cabinet* (DEC) dengan mengkombinasikan *InterPlanetary File System* (IPFS) dan *Hyperledger Fabric* (HF) sebagai sistem penyimpanan yang dapat didistribusikan. Dengan mengusulkan pendekatan alternatif IPFSChain model, dimungkinkan untuk mencapai kemudahan transfer data, kepercayaan data yang lebih baik dan perlindungan kepemilikannya. Kontribusi dari penelitian ini adalah memberikan konsep IPFSChain sebagai model penyimpanan terdistribusi dan semua aktivitas pada aset dapat diaudit dengan baik dengan mempertimbangkan kaidah CoC.

Kata kunci

Bukti Digital, CoC, DEC, IPFS, *Hyperledger Fabric,* IPFSChain

Abstract

InterPlanetary File System based Hyperledger Fabric on Digital Evidence Cabinet for Digital Evidence Management

Dematerialization of physical evidence is an asset of digitizing information, where information from physical evidence becomes more important than physical evidence itself. In principle, to preserve digital evidence, an accurate and reliable distributed storage system is needed that is packaged in terms of asset digitization which are summarized in chain of custody (CoC) documents. In addition to handling all the needs related to the storage system, one thing that is no less important is to focus on the ease of distributing evidence on the network safely and reliably. This is one of the most important parts of interpreting the effectiveness of digital forensic investigations. However, several problems arose regarding the concept of digital evidence asset storage management, which still cannot be distributed, and are difficult to track. The purpose of this study is to complement the concept of Digital Evidence Cabinet (DEC) by combining InterPlanetary File System (IPFS) and Hyperledger Fabric (HF) as a distributable storage system. By proposing an alternative approach to the IPFSChain model, it is possible to achieve ease of data transfer, better data trust and protection of its ownership. The contribution of this research is to provide the concept of IPFSChain as a distributed storage model and all activities on assets can be audited properly by considering the rules of CoC.

Keywords

Digtial Evidence, CoC, DEC, IPFS, Hyperledger Fabric, IPFSChain

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Mei 2022

Jefrul Hanafi, S.Kom.

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Publikasi berikut menjadi bagian dari Bab 3

Kontributor	Jenis Kontribusi
Jefrul Hanafi	Mendesain eksperimen (70%) Menulis *paper* (75%)
Yudi Prayudi	Mendesain eksperimen (20%) Menulis dan mengedit *paper* (5%)
Ahmad Luthfi	Mendesain eksperimen (10%) Menulis dan mengedit *paper* (20%)

Halaman Kontribusi

Penelitian ini tidak terlepas dari berbagai saran maupun bimbingan dan berbagai pihak, mulai dari penelitian, seminar proposal, seminar progress, hingga seminar pendadaran. Pihak-pihak tersebut, antara lain, Dr. Yudi Prayudi, S.Si., M.Kom dan Ahmad Luthfi, S.Kom., M.Kom., Ph.D.

Halaman Persembahan

Bismillahirrahmanirrahim.

Alhamdulillah atas ridho Allah Subhanahu Wa Ta'ala karya ini saya persembahkan kepada kedua orang tua dan kawan-kawan yang telah mendukung, memberikan semangat dan motivasi dalam menyelesaikan pendidikan magister komputer saya ini, secara khususnya kepada:

1. Ayahanda (Afdal) dan Ibunda (Azwati (Almh.)) yang selalu mendo'akan anaknya menjadi orang yang berguna seperti yang diharapkan, baik bangsa, dan tentunya agama. Dan yang terpenting adalah pengorbanan yang tak lekang oleh masa dan tak terbalas oleh upaya.
2. Abang (Azharianda) dan Kakak (Susri Dewi) yang selalu mendukung baik dari segi moril maupun materil. Dan yang terpenting adalah saling mengingatkan dan menjaga dalam kekompakkan.
3. Dea Putri Ananda yang menjadi penyemangat dalam dunia pendidikan dan tentunya masa depan. InsyaAllah, yang terpenting adalah calon ibu yang salihah buat anak-anak saya yang salih-salihah (Aamiin).
4. Tidak lupa tentunya kepada kawan-kawan seperjuangan Forensika Digital '18 yang memberikan dukungan selama menempuh pendidikan ini.

Kata Pengantar

Assalamu'alaikum Warohmatullahi Wabarokatuh.

Puji syukur penulis haturkan kepada Allah SWT atas limpahan rahmat dan karunia yang diberikan kepada penulis sehingga dapat menyelesaikan laporan penelitian tesis yang berjudul "*InterPlanetary File System* pada *Digital Evidence Cabinet* berbasis *Hyperledger Fabric* untuk Manajemen Bukti Digital". Adapun maksud dari penulisan laporan penelitian ini adalah sebagai persyaratan dalam mencapai jenjang pendidikan Magister Informatika konsentrasi Forensika Digital di Fakultas Teknologi Industri, Universitas Islam Indonesia. Dalam proses penyelesaian tesis ini penulis tidak dapat menyelesaikannya bila tidak ada turut serta pihak lain yang juga ikut membantu baik secara langsung maupun tidak langsung. Untuk itu penulis ingin menyampaikan rasa terima kasih kepada beberapa pihak yang telah mendukung dalam penyusunan tesis ini, antara lain:

1. Bapak Fathul Wahid, S.T., M.Sc., Ph.D, selaku rektor Universitas Islam Indonesia yang memberikan kesempatan kepada penulis untuk menimbah ilmu di Universitas Islam Indonesia.
2. Bapak Prof. Hari Purnomo, M.T selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia yang memberikan fasilitas dan bantuan untuk belajar.
3. Ibu Izzati Muhimmah, ST., M.Sc., Ph.D, selaku Ketua Program Studi InformatikaProgram Magister Fakultas Teknologi Industri, Universitas Islam Indonesia, yang selalu memberikan semangat kepada setiap mahasiswa agar segera menyelesaikan tesis.
4. Bapak Dr. Yudi Prayudi, S.SI., M.Kom, dan Bapak Ahmad Luthfi, S.Kom., M.Kom., Ph.D. selaku dosen pembimbing yang telah banyak meluangkan waktunya dalam memberikan berbagai saran selama proses bimbingan.
5. Seluruh Dosen, staff administrasi dan civitas Magister Informatika Universitas Islam Indonesia, baik secara langsung maupun tidak langsung telah membantu penulis selama masa studi penulis.
6. Seluruh keluarga baik Bapak, Ibu, Abang, dan Kakak yang telah mencurahkan segenap cinta, kasih sayang, perhatian dan dukungan baik moril maupun materil.
7. Rekan-rekan mahasiswa MI khususnya konsentrasi Forensika Digital angkatan 18 yang selama ini berjuang bersama dan selalu memberikan semangat satu sama lain. dalam penyusunan laporan tesis.

Adapun yang masih ada kekurangan maupun kesalahan dalam laporan ini, penulis membuka lebar kritik dan sarannya agar dapat diperbaiki supaya kedepannya lebih maksimal sesuai yang diinginkan. Akhir kata dari penulis, terimakasih sebesar-besarnya semoga laporan ini bermanfaat dan bisa membantu rekan-rekan yang membutuhkan khususnya mahasiswa/mahasiswi Universitas Islam Indonesia.

Wassalamu'alaikum Warahmatullahi Wabarokatuh.

Daftar Isi

Daftar Tabel

Daftar Gambar

Glosarium

APH	- Aparat Penegak Hukum
B-CoC	- Blockchain Chain of Custody
B-DEC	- Blockchain Digital Evidence Cabinet
BNA	- Business Network Archive
CoC	- Chain of Custody
D-CoC	- Digital Chain of Custody
DEC	- Digital Evidence Cabinet
DLT	- Distributed Ledger Technology
HF	- Hyperledger Fabric
IPFS	- InterPlanetary File System
MSC	- Multy Smart Contract
TKP	- Tempat Kejadian Perkara

BAB 1
Pendahuluan

1.1 Latar Belakang

Mayoritas kejahatan saat ini memiliki komponen digital, sehingga APH diwajibkan oleh undang-undang untuk menyimpan bukti digital untuk sejarah suatu kasus (Granja & Rafael, 2017). Selain itu, volume data untuk investigasi kejahatan dunia maya terus tumbuh pada tingkat yang belum pernah terjadi sebelumnya dan menciptakan kebingungan bagi lembaga penegak hukum (Kao et al., 2019). Ini juga dapat berdampak pada pencatatan dan dokumentasi serta kompleksitas pada pengelolaan bukti digital (Prayudi et al., 2014). DEC dibangun sebagai alternatif dari tantangan tersebut. Namun manajemen dari DEC sebagai mekanisme penyimpanan bukti digital dan CoC terdapat kelemahan, salah satunya adalah sistem penyimpanan yang bersifat terpusat. Dimana file rentan akan modifikasi dan hilangnya dokumen CoC. Selain itu, dengan adanya *third-party* (pihak ketiga, yaitu penyedia jasa *cloud computing*) pada sistem terpusat juga dapat mengurangi tingkat kepercayaan dan transparansi pada data.

B-CoC memberikan solusi atas persoalan pada DEC, dimana arsitektur berbasis *blockchain Ethereum* untuk mendematerialisasikan proses CoC dalam forensik digital yang dilakukan dalam penelitian (Bonomi et al., 2018). Konsep tersebut memberikan keamanan dari kehilangan dan perubahan pada data, karena *blockchain* bersifat *distributable* dan terenkripsi. Namun konsep ini memiliki kelemahan yaitu tidak direkomendasikan ketika satu bukti digital dikerjakan oleh banyak investigator. Dengan kata lain, satu investigator hanya dapat mengakses CoC untuk satu bukti digital. Solusi untuk masalah tersebut dilakukan penelitian lanjut oleh (Yunianto et al., 2019) dengan B-DEC, dan juga (Lone & Mir, 2019) *Forensic-chain.* Namun kedua konsep dari model tersebut menurut penelitian yang dilakukan oleh (Putra & Prayudi, 2021) terdapat kekurangan yang sama, yaitu pada penggunaan satu *smart contract.* Dalam penelitiannya menyatakan bahwa, menggunakan hanya dengan satu *smart contract* integritas dan pengamanan bukti digital dianggap masih kurang. Sehingga dikembangkan lanjut dengan menggunakan *multi smart contract* yang berbasis *naive chain.*

Multi smart contract pada bukti digital dan CoC berbasis *naive chain* yang diteliti oleh (Putra & Prayudi, 2021) ini sebagai solusi dari masalah penggunaan yang hanya satu

smart contract dan masih bersifat general serta bukti digital tidak dikelompokkan berdasarkan jenis. Model *multi smart contract* ini memberikan integritas yang tinggi pada bukti digital. Karena semua informasi dari bukti digital tersebut diekstrak menggunakan GetID3 kemudian disimpan ke dalam sistem. Namun, kurangnya efisiensi dari model ini serta berdampak pada kinerja dan volume sistem penyimpanan. Seharusnya, informasi yang disimpan ke dalam sistem *blockchain* yang berupa metadata tersebut cukup mengikuti kaidah atau standar informasi dari CoC.

Di sisi lain, penelitian yang dilakukan oleh (Nyaletey et al., 2019) tentang menciptakan jejak audit yang jelas. Dengan mengusulkan pendekatan baru yaitu BlockIPFS untuk mencapai kepercayaan yang lebih baik dari data dan perlindungan kepengarangan. Penggabungan sistem IPFS dan teknologi *blockchain* HF yang terintegrasi, tentunya memberikan keamanan pada data dan volume penyimpanan yang juga besar. Namun Nyaletey sendiri mengatakan bahwa konsep ini memberikan *overhead* yang menghabiskan sumber daya dan waktu komputasi.

Dua persoalan di atas, dilakukan pendekatan baru yaitu IPFSChain model. Pendekatan ini akan dilakukan pengembangan model alternatif menggunakan *blockchain* HF sebagai *on-chain* dan IPFS sebagai *off-chain*. Model ini dimungkinkan untuk kemudahan transfer data aset, tingkat kepercayaan yang lebih baik, dan perlindungan kepemilikannya. Dimana IPFSChain ini cukup menggunakan satu *chaincode* atau *smart contract* dan bukti digital dikelompokkan berdasarkan kasus. Adapun informasi yang dientri ke dalam CoC berdasarkan pada standar informasi CoC yang diteliti oleh (Prayudi et al., 2014) dan otoritas hak akses tiap peserta berdasarkan peran dari masing-masing peserta. Pendekatan pada IPFSChain model ini melalui konsep *Digital Evidence Management Framework, distributed storage* dan akses kontrol, serta keamanan transaksi yang teraudit. Pendekatan pada HF adalah menggunakan modular *Hyperledger Composer* yang memungkinkan ekstensibilitas dan fleksibilitas. Sedangkan pendekatan pada IPFS melalui optimasi dan kapasitas penyimpanan. *Output* dari penelitian ini adalah *logs* aktivitas yang dilakukan oleh admin maupun peserta atau *stakeholders* (*first responder, investigator, officer,* dan aparat penegak hukum). Dengan kata lain, *logs* tersebut sebagai rantai jejak yang terekam dari awal sampai dengan akhir aktivitas secara transparan atau dapat dilihat oleh semua peserta yang tergabung dalam jaringan.

1.2 Rumusan Masalah

Dari pemaparan latar belakang tersebut dapat dirumuskan beberapa masalah yaitu:

1. Bagaimana perancangan konsep DEC pada IPFSChain model terhadap konten bukti digital dan CoC sehingga mudah dan aman untuk akses dan transfer data dalam jaringan?
2. Bagaimana implementasi konsep DEC pada IPFSChain model terhadap konten bukti digital dan CoC sehingga mudah dan aman untuk akses dan transfer data dalam jaringan?
3. Bagaimana menguji rancangan IPFSChain model?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu:

1. Membangun konsep IPFSChain model yang terdistribusi untuk kemudahan akses dan transfer data dalam jaringan.
2. Mengetahui seberapa besar tingkat keberhasilan konsep IPFSChain model dapat diterapkan.
3. Mengetahui seberapa efektif dan efisien konsep IPFSChain model.

1.4 Manfaat Penelitian

Berikut manfaat dari penelitian ini adalah:

1. Memberikan kemudahan dan keamanan pada akses dan transfer data dalam jaringan IPFSChain model.
2. Implementasi IPFSChain model diharapkan dapat memberikan tingkat kepercayaan pada konten bukti digital dan CoC secara transparansi serta semua aktivitas dapat diaudit.
3. Dengan pengelompokan bukti digital berdasarkan kasus diharapkan dapat memberikan efektivitas dan efisiensi pada tugas *stakeholders* (investigator, *digital evidence analyst*, ataupun aparat penegak hukum).

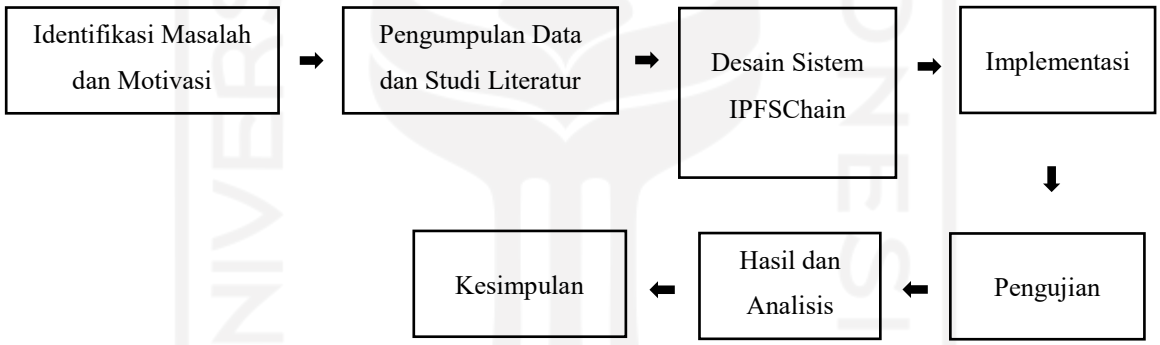
1.5 Batasan Masalah

Adapun beberapa batasan masalah yang akan diteliti dari penelitian ini adalah:

1. Fokus penelitian pada konsep DEC berbasis HF.
2. Metadata bukti digital didokumentasikan ke dalam CoC berdasarkan standar informasi CoC dari DEC.
3. Format bukti digital yang digunakan adalah .txt, .jpg, .pdf, .mp3, dan .mp4 serta bukti digital dikelompokkan hanya berdasarkan kasus.

1.6 Metodologi Penelitian

Dalam penelitian ini, metodologi yang digunakan terdiri dari 8 tahapan yaitu identifikasi masalah dan motivasi, pengumpulan data dan studi literatur, desain dan pengembangan, implementasi, simulasi dan skenario kasus, pengujian, analisis, kesimpulan. Adapun uji coba pada penelitian ini adalah dengan menggunakan metode eksperimen. Berikut alur diagram pada metodologi yang digunakan pada penelitian ini:



1.7 Sistematika Penulisan

Berikut adalah sistematika yang dilakukan pada penelitian ini:

BAB 1 Pendahuluan

Bab ini menjelaskan terkait bagian penelitian yang diteliti. Didalamnya terdapat temuan beberapa masalah baik masalah saat ini maupun masalah dari penelitian sebelumnya. Selain itu terdapat juga rumusan masalah, alasan penelitian serta kontribusi dari penelitian. Dan didalamnya termaktub sistematika penulisan dari penelitian ini.

BAB 2 Tinjauan Pustaka

Bab ini menjelaskan kajian pustaka dengan bidang penelitian yang relevan dan rumusan masalah, serta ditutup dengan sebuah ringkasan hasil penelitian sebelumnya

yang akan dikembangkan lebih lanjut. Ringkasan dapat direpresentasikan dalam bentuk gambar, model, atau tabel.

BAB 3 Metode Penelitian

Bagian ini terdapat argumentasi atas metode yang dipilih beserta rincian langkah-langkahnya. Metode yang dipilih guna untuk pengumpulan, pengujian, dan analisa data. Selain itu termuat pula hipotesis yang diformulasi dan model yang dikonstruksikan.

BAB 4 Hasil dan Pembahasan

Pada bab ini memaparkan deskripsi sistem yang telah dibuat, konsideran munculnya rancangan, informasi dari data yang dikumpulkan (dari observasi, simulasi, kinerja sistem, skenario dari proses pengembangan, dokumen, dan lain-lain).

BAB 5 Kesimpulan dan Saran

Bab ini menyimpulkan masalah, pokok temuan, dan diskusi yang ditulis di kajian pustaka. Bagian ini juga memuat bagaimana temuan penelitian menjawab rumusan masalah, implikasi hasil penelitian, disabilitas, dan arah penelitian selanjutnya.

BAB 2
Tinjauan Pustaka

2.1 *Chain of Custody* (CoC)

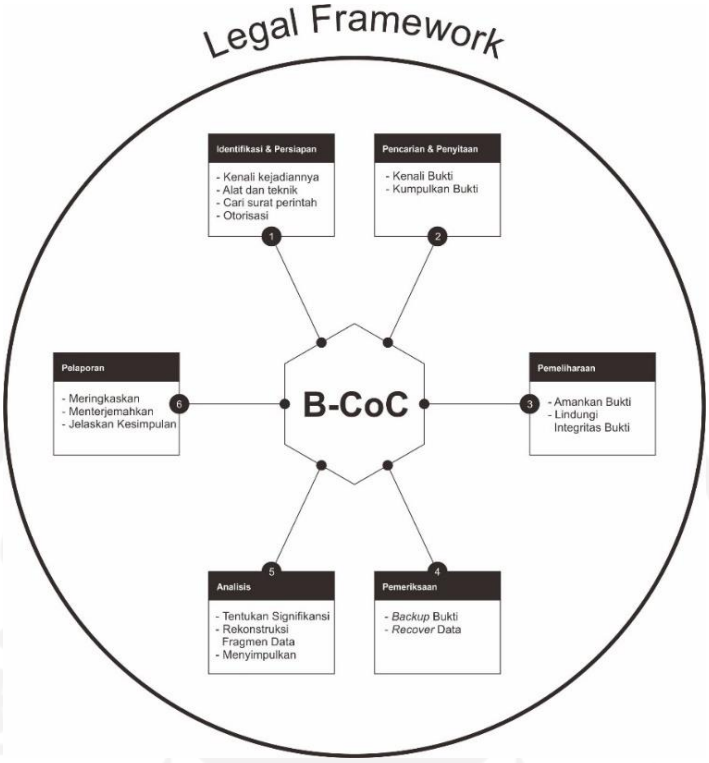
CoC dapat didefinisikan sebagai suatu proses yang digunakan untuk memelihara dan mendokumentasikan sejarah kronologis penanganan bukti digital (Giova, 2011). Menurut (Bonomi et al., 2018) CoC adalah proses validasi bagaimana semua bukti telah dikumpulkan, dilacak dan dilindungi dalam perjalanannya ke pengadilan. Dengan kata lain, CoC dapat diartikan sebagai manajemen pelestarian dokumen bukti digital.

Menurut Badan Standar Nasional (2014) *form* CoC paling tidak memiliki beberapa syarat, yaitu:

1. Mengidentifikasi unik pada barang bukti
2. Data siapa yang mengakses barang bukti beserta dengan waktu dan lokasinya
3. Data siapa yang melakukan pengecekan terhadap keluar masuknya barang bukti dari tempat penyimpanan jika memang itu terjadi
4. Alasan kenapa barang bukti dikeluarkan dan otoritas pihak berwenang
5. Perubahan terhadap barang bukti yang tidak terelakkan dengan menyertakan alasan dan siapa yang bertanggung jawab.

CoC terdiri dari dokumen bukti dalam bentuk fisik (perangkat) dan dokumen bukti dalam bentuk digital. Terdapat beberapa perbedaan yaitu, pada bukti fisik CoC lebih pada integritas bukti agar tidak terkontaminasi oleh orang yang tidak berhak. Pada bukti digital dibutuhkan perlakuan lebih karena pada kenyataannya tidak dapat dimasukkan ke dalam kantong bukti (*evidence bag*) layaknya bukti fisik kemudian dikelompokkan dalam kabinet (Prayudi & SN, 2015)

Investigator digital forensik diharapkan kompeten dalam penggunaan berbagai alat forensik dan memastikan bahwa setiap proses investigasi dilakukan dalam kerangka hukum yang dapat diterima dari sistem pengadilan. Oleh karena itu dalam penelitian kali ini mencoba mengacu pada beberapa model yang diusulkan oleh peneliti sebelumnya (Ami-Narh & Williams, 2008) terkait kerangka hukum dalam proses penyitaan dan penanganan bukti digital, yakni diantaranya dapat dilihat pada gambar 3.2.



Gambar 2.1 Model proses penyitaan dan penanganan bukti digital forensik. (Ami-Narh & Williams, 2008).

Bidang-bidang yang diidentifikasi yang paling mungkin dipertanyakan dalam kasus hukum adalah kasus yurisdiksi, pencarian dan penyitaan, spoliasi bukti, pelestarian bukti, pemeriksaan dan analisis bukti (Ami-Narh & Williams, 2008). Sehingga salah satu poin penting yang dipertanyakan yaitu pelestarian bukti. Dari gambar 2.1 tersebut diharapkan dapat memberikan gambaran umum terkait CoC yang berbasiskan blockchain.

2.2 *Blockchain*

Menurut (Dauglas & Lancaster, 2018) *blockchain* hanyalah metode untuk membuat catatan transaksi melalui ideologi *peer to peer*, sebagai istilah sederhana untuk berbagi dan menyusun data dalam jaringan. *Blockchain* lebih dari sekedar *cryptocurrency*. *Blockchain* pertama yang mereka buat adalah untuk membantu memverifikasi transaksi aset tertentu. Semua *blockchain* ini pada dasarnya memiliki kode dasar yang sama yang menciptakan catatan permanen dari transaksi yang dapat dilacak dan diverifikasi.

Blockchain merupakan teknologi terkini yang pada dasarnya mempunyai *ledger* (database) yang didistribusikan secara terbuka dan mencatat seluruh transaksi yang ada secara detail pada sebuah *block.* Dimana setiap *block* mempunyai waktu dan saling terhubung ke blok sebelumnya dan tahan terhadap modifikasi (Mougayar, 2016). *Blockchain*

diperkenalkan sebagai teknologi yang digunakan pada Bitcoin oleh Satoshi Nakamoto di jurnalnya pada rentang tahun 2008/2009 (Shoris, 2018). Namun *blockchain* tidak terbatas pada aktivitas merekam transaksi keuangan saja (Singhal et al., 2018).

Konsep teknis pada *blockchain* menurut IBM (2018) memiliki beberapa istilah seperti:

1. *Peers* adalah layanan jaringan yang melayani dan memelihara *ledger state* dan dapat menjalankan *smart contracts.*
2. *Channels* merupakan bagian dari jaringan yang dapat membagikan sebuah *ledger*.
3. *Certificate Authorities* menyediakan layanan identitas pada setiap partisipan dalam sebuah jaringan.
4. *Smart Contracts* merupakan suatu mekanisme yang mengatur logika transaksi dimana setiap *output* membutuhkan persetujuan *peer* pada jaringan.
5. *Consensus* adalah proses yang merupakan persetujuan yang dilakukan dalam sebuah jaringan *peer*.
6. *Ordering Service* merupakan bagian dari transaksi dan *block* terdistribusi pada *peers.*

2.3 *Hyperledger Fabric* (HF)

Hyperledger Fabric (2019) menjelaskan bahwa HF adalah *platform Distributed Ledger Technology* (DLT) *open source* yang diizinkan perusahaan, dirancang untuk digunakan dalam konteks perusahaan, yang memberikan beberapa kemampuan membedakan kunci dibandingkan *platform ledger* atau *blockchain* populer lainnya. *Fabric* adalah *platform ledger* terdistribusi pertama yang mendukung *smart contract* yang ditulis dalam bahasa pemrograman java, go dan node.js. *Platform fabric* para peserta dalam jaringannya dapat diketahui satu sama lain karena bersifat privat (*permissioned*). Sedangkan yang bersifat publik yang mana para pesertanya adalah anonim oleh karena itu mungkin tidak percaya satu sama lain, misalnya menjadi pesaing dalam industri yang sama.

Salah satu yang terpenting pembeda *platform* adalah dukungannya untuk protokol konsensus *pluggable* yang memungkinkan *platform* untuk lebih efektif disesuaikan dengan kasus penggunaan tertentu dan model kepercayaan. *Fabric* dapat memanfaatkan protokol konsensus yang tidak memerlukan *cryptocurrency* asli atau monetisasi data untuk insentif penambangan yang mahal atau untuk memicu eksekusi *smart contract* (Hyperledger Fabric, 2019; Aran Davies, DevTeam.Space, 2018). Menghindari mata uang kripto berarti mengurangi beberapa resiko atau serangan vektor yang signifikan. Dan juga dapat diartikan bahwa *platform* dapat dikerahkan dengan biaya operasional yang sama seperti sistem

terdistribusi lainnya. *Hyperledger framework* memiliki beberapa perbedaan tentunya dengan teknologi blockchain yang bersifat publik atau tanpa izin ("*permission-less*"). Berikut penjelasannya pada tabel dibawah ini.

Tabel 2.1 Perbedaan hyperledger (permission) dengan blockchain tanpa izin (permission-less). Plessing, P. (2019).

	Bitcoin	***Ethereum***	***Hyperledger Frameworks***
Cryptocurrencybased	Yes	Yes	No
Permissioned	No	No	Yes
Pseudo-anonymous	Yes	No	No
Auditable	Yes	Yes	Yes
Immutable-ledger	Yes	Yes	Yes
Modularity	No	No	Yes
Smart contracts	No	Yes	Yes
Consensus protocol	PoW	Pow	Varios**

Tabel 2.2 Diferensiasi *permissionless, permissioned,* dan *central database*. Wust & Gervais (2017).

	Permissionless **(Tanpa Izin)**	***Permissioned*** **(Diizinkan)**	***Central Database*** **(Database Pusat)**
Hasil	Rendah	Tinggi	Sangat Tinggi
Latensi	Lambat	Medium	Cepat
Jumlah Pembaca	Tinggi	Tinggi	Tinggi
Jumlah Penulis	Tinggi	Lambat	Tinggi
Jumlah Penulis yang tidak dipercaya	Tinggi	Lambat	0
Mekanisme Konsensus	Terutama PoW, beberapa PoS	BFT Protocols (e. g. PBFT)	Tidak ada
Dikelola secara terpusat	Tidak	Ya	Ya

Konsep HF menurut (Benhamouda et al., 2019) yang terdir dari *Block (Data Model Layer), Nodes/Blockchain (Execution Layer), Consensus* (Lapisan Konsensus), dan *Distributed Ledger Technology* (DLT) (Lapisan Konsensus), yaitu node yang memiliki akses ke buku besar disebut rekan, dan masing-masing rekan memiliki organisasi. Ada dua fase jika terjadinya proses transaksi ke *fabric* yaitu pertama, klien yang meminta transaksi

pertama kali mendekati satu atau lebih rekan dengan proposal transaksi dan meminta mereka untuk mengeksekusi dan mendukung proposal tersebut. Kedua, *peer* yang mendukung proposal tersebut kemudian menjalankan kontrak pintar dalam *fabric* disebut *chaincode*.

Tujuannya adalah untuk menentukan apakah akan mendukung transaksi atau tidak. Lalu bagaimana transaksi ini mengubah status pada buku besar. Setelah diperoleh cukup banyak dukungan, klien mengirimkan transaksi yang didukung ke layanan pemesanan yang membebankan pesanan linear pada transaksi dan kemudian ditambahkan ke buku besar. Biasanya, buku besar hanya memiliki satu kebijakan dukungan yang berlaku untuk semua transaksi di dalamnya.

2.4 *InterPlanetary File System* (IPFS)

IPFS adalah sistem file terdistribusi *peer-to-peer* yang berusaha menghubungkan semua perangkat komputasi dengan sistem file yang sama (Benet, 2014). Dengan kata lain, IPFS adalah sistem penyimpanan terdistribusi yang mirip dengan BitTorrent. Data dipecah menjadi beberapa bagian dan didistribusikan ke seluruh node dalam jaringan. Elemen data diberi *hash* IPFS yang unik, sebagai pengenal untuk file tersebut. Sehingga file tersebut dapat diakses melalui hash IPFS. Apabila seseorang ingin mengakses suatu file tetapi tidak mengetahui nilai *hash* dari file tersebut, maka file itu tidak dapat diakses. Saat file diunggah ke jaringan, *hash* IPFS unik dari file tersebut dibuat dan diunggah ke jaringan *blockchain*. Bersamaan dengan itu, metadata dari file tersebut seperti informasi penulis, ukuran file, jenis file yang diunggah dan stempel waktu (Rahalkar & Gujar, 2019).

Tulang punggung IPFS bergantung pada protokol *Data Hash Table* (DHT). Ini adalah penyimpanan nilai kunci yang menggunakan teknologi terdistribusi untuk menyimpan data. Kerugian DHT adalah integritas dan privasi data. Karena setiap node tidak memiliki salinan dari semu data yang disimpan di jaringan, waktu henti node tertentu menyebabkan hilangnya data (Rahalkar & Gujar, 2019).

Berdasarkan penjelasan oleh (Benet, 2014) dalam papernya, bahwa IPFS adalah sistem file terdistribusi yang mensintesis sukses ide-ide penting dari sistem *peer-to-peer* sebelumnya, termasuk DHT, BitTorrent, Git, dan SFS. Kontribusi IPFS adalah menyederhanakan, mengembangkan, dan menghubungkan teknik yang telah terbukti menjadi satu sistem kohesif, lebih besar daripada jumlah bagian-bagiannya. Protokol IPFS dibagi menjadi tumpukan sub-protokol yang bertanggung jawab untuk fungsi yang berbeda yaitu:

1. *Identities*: mengelola pembuatan dan verifikasi identitas node.

2. *Network*: mengatur koneksi ke *peer* lain, menggunakan berbagai protokol jaringan yang mendasarinya.
3. *Routing*: memelihara informasi untuk menemukan peer dan objek tertentu. Menanggapi pertanyaan lokal dan pertanyaan jarak jauh. Secara default menggunakan DHT, tetapi dapat ditukar.
4. *Exchange*: protokol pertukaran blok baru (BitSwap) yang mengatur distribusi blok yang efisien. Dimodelkan sebagai pasar, dengan lemah memberikan insentif bagi replikasi data. Strategi Perdagangan dapat ditukar.
5. *Object*: Merkle DAG objek tidak dapat diubah yang dialamatkan dengan konten dengan tautan. Digunakan untuk merepresentasikan struktur data arbitrer, mis. hirarki file dan sistem komunikasi.
6. *Files*: hirarki sistem file berversi yang terinspirasi oleh Git.
7. *Naming*: Sistem nama yang dapat berubah dengan sendirinya.

Tabel 2.3 Ulasan Kritis

No	Peneliti	Masalah dan Tujuan	Metode	Hasil
1	(Giova, 2011)	Masalah: Kompleksitas jenis bukti digital membuat semakin kompleksnya kebutuhan akan dokumen *chain of custody.* Dimana dokumen *chain of custody* tidak efisien dan tidak dapat menjamin proses legal formal. Tujuan: Membangun sistem yang dapat membangun akomodasi *chain of custody* pada studi kasus file AFF (*Advance Forensic Format*)	Menggunakan RDF sebagai generator file XML guna mencatat semua aktivitas pada sebuah bukti digital dengan jenis AFF	Sistem RDF (*Resource Description Framework*) yang menghasilkan *chain of custody* pada bukti digital dengan format AFF
2	(Prayudi et al., 2014)	Masalah: Penanganan *chain of custody* bukti digital sama dengan bukti fisik. Padahal bukti digital mempunyai kompleksitas dan kebutuhan lebih. Tujuan: Membangun dan mengimplementasikan kantong barang bukti pada bukti digital	Mendesain model *Digital Evidence Cabinet* sesuai kebutuhan *chain of custody* untuk bukti digital	*Framework Digital Evidence Cabinet* yang dapat menangani bukti digital sebagai kantong barang bukti dan kabinet pada bukti fisik
3	(Lone & Mir, 2017)	Masalah: Integritas bukti digital dan penanganannya pada *chain of custody* menjadi permasalahan ketika banyak jenis bukti digital tanpa diimbangi dengan skema dan metode dokumentasinya. Tujuan: Menerapkan teknologi *blockchain* untuk menyimpan aktivitas *chain of custody*	Mendesain sistem yang dapat mengimplementasikan teknologi *blockchain* sebagai media penyimpanan dokumen *chain of custody*	*Forensic-chain* sebagai model penerapan *blockchain* pada *chain of custody* dengan menggunakan Ethereum dan *Smart Contract*
4	(Ratnasari et al., 2018)	Masalah:	Menggunakan dan mendesain XML pada *chain of custody* bukti	Dokumen XML sebagai pendamping bukti digital yang didalamnya terdapat

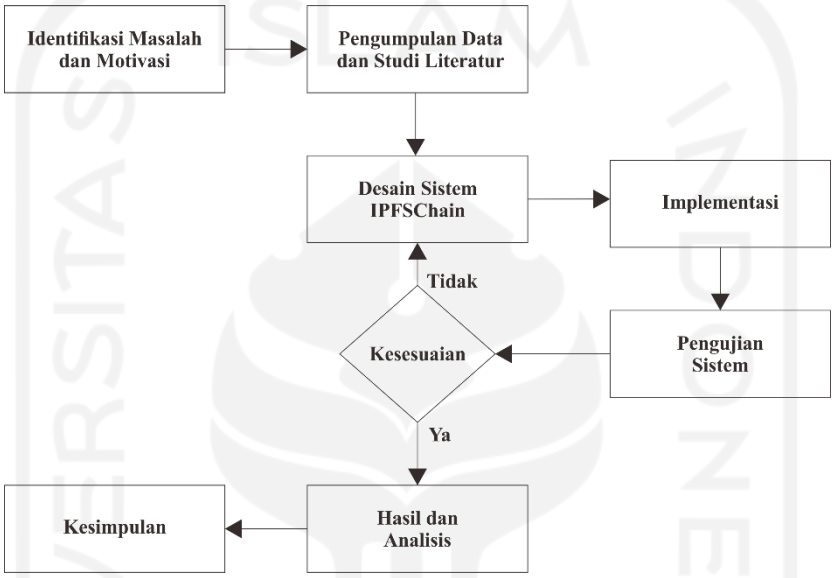
		Penggunaan dokumen *chain of custody* pada bukti digital dapat menjadi permasalahan karena berbeda karakteristik. Tujuan: Membangun pendekatan XML untuk membuat dokumen *chain of custody* bukti digital.	digital yang berasal dari metadata barang bukti digital.	beberapa informasi yang di *generate* dari bukti digital yang memuat informasi terhadap bukti digital tersebut
5	(Bonomi et al., 2018)	Masalah: Persyaratan proses CoC dipenuhi dengan melakukan penyerahan bukti fisik dimana pada setiap tahap, dokumen diisi dan ditandatangani di depan petugas. Tujuan: Proses dematerialisasi CoC bukti digital yang berbasis blockchain ethereum untuk melacak perubahan kepemilikan selama siklus hidup bukti digital.	Chain of Custody (B-CoC) berbasis Blockchain yang didasarkan pada blockchain publik. Dalam hal ini adalah etherium.	B-CoC terbukti menjadi dukungan yang efektif untuk proses CoC karena mampu mempertahankan beban kerja yang realistis dengan *overhead* yang dapat diterima dalam hal memori yang digunakan untuk menyimpan rantai.
6	(Yunianto et al., 2019)	Masalah: Tidak menunjukkan design informasi yang disimpan dan juga informasi berkaitan dengan pihak yang mengakses bukti digital. Tujuan: Membangun perancangan *digital evidence cabinet* (DEC) pada *chain of custody* dengan menggunakan teknologi *blockchain.*	Menggunakan Ethereum. Ethereum yang digunakan adalah jenis *private* yaitu *Proof of Authority* (PoA) atau *Istanbul Byzantine Fault Tolerance* (IPBF) dengan pendekatan Bukti Digtal Cabinet (B-DEC)	Aplikasi B-DEC (*Blockchain Digital Evidence Cabinet*) sebagai manajemen bukti digital dengan memastikan otentikasi dokumen *chain of custody* dan kontrol akses pengguna pada bukti digital.
7	(Lone & Mir, 2019)	Masalah: Jaringan Ethereum adalah tentang anonimitas (pseudo-anonimitas) di mana setiap orang dapat melihat transaksi tetapi hampir tidak mungkin untuk menyimpulkan siapa yang terlibat dalam transaksi ini. Aset (Bukti Digital) perlu dicegah agar tidak dirusak oleh peserta yang tidak dipercaya. Tujuan:	*Forensic-Chain* yang berbasis *Hyperledger Composer* dengan *Proof of Concept* (PoC)	Model *Forensic-Chain* yang berbasis Hyperledger Composer dimana informasi bukti hanya terbatas pada peserta yang merupakan bagian dari *blockchain* dan disahkan oleh rekan admin yang dimiliki oleh organisasi konsorsium.

		Membangun sistem dimana informasi tentang bukti digital hanya diketahui atau dapat diakses oleh peserta yang disahkan.		
8	(Nyaletey et al., 2019)	Masalah: (Lone & Mir, 2019) menyatakan bahwa keterbatasan pada skala penyimpanan. Tujuan: Untuk melengkapi IPFS dengan teknologi blockchain, dengan mengusulkan pendekatan baru (BlockIPFS) untuk menciptakan jejak audit yang jelas.	IPFS dan HF yang terintegrasi	Mampu melacak semua jejak aktivitas pada sistem IPFS dengan mengintegrasikan HF, serta akses yang dikendalikan.
9	(Putra & Prayudi, 2021)	Masalah: penggunaan satu *multi smart contract* dan belum ada pengelompokkan pada jenis bukti digital. Tujuan: Menerpakan metode *multi smart contract* pada *Naive Chain* untuk meningkatkan integritas bukti digital dan *chain of custody*.	*Multi smart contract* berbasis *Naive chain*	Sebuah model *blockchain* yang bisa memisahkan *chain of custody* berdasarkan jenis atau tipe filenya dalam *smart contract* yang berbeda-beda.
10	Usulan penelitian	Masalah: (Nyaletey et al., 2019) menyatakan bahwa IPFS dan HF jika diintegrasikan maka dapat memberikan *overhead* yang menghabiskan sumber daya dan waktu komputasi. (Putra & Prayudi, 2021) kurangnya efektivitas dan efisiensi jika semua informasi dari bukti digital diekstrak dan disimpan ke dalam sistem *naive chain.* Tujuan: Mengimplementasikan IPFS dan *Hyperledger Fabric* dengan konsep *on-chain* dan *off-chain* pada konten bukti digtal dan metadata *chain of custody* untuk kemudahan dan keamanan akses dan transfer aset dalam jaringan.	Manajemen bukti digital dengan konsep *on-chain* dan *off-chain.*	IPFSChain model yang memberikan transparansi, kemudahan, dan keamanan untuk akses dan transfer aset.

BAB 3
Metodologi

3.1 Alur Metodologi Penelitian

Penelitian ini secara umum terdiri dari 8 tahapan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur metode penelitian.

Metode penelitian ini berdasarkan *review* yang terkait dari penelitian sebelumnya untuk mengembangkan ide penelitian yang sudah ada. Selanjutnya dilakukan analisis untuk mengetahui prosedur apa saja yang perlu diterapkan pada penggunaan metode yang sesuai dalam proses pemeliharaan bukti digital, supaya otentisitas bukti digital lebih terjamin. Untuk menghindari jebakan spoliasi atau rampasan pada bukti digital, investigator harus memastikan perlindungan bukti selama investigasi, mencegah masuknya virus komputer, dan mengelola dengan baik bukti yang relevan dan CoC yang berkelanjutan (Leeds & Marra, 2000). Peneliti forensik dapat memastikan pelestarian bukti digital sesuai dengan persyaratan hukum dengan mengambil tindakan berikut (Thomas, 2004; Hershensohn, 2005):

1. Hindari sumber magnetis, kelembaban, panas berlebih, atau sangat dingin, goncangan, dan lain-lain.
2. Bukti yang dihapus harus didokumentasikan dan disegel.

3. Dokumentasikan seluruh proses transportasi bukti.
4. Tangani bukti dan peralatan elektronik dengan benar untuk menghindari kerusakan.
5. Menjaga CoC.

3.2 Identifikasi Masalah dan Motivasi

Identifikasi masalah adalah upaya eksplorasi pada masalah manajemen CoC yang bersifat digitalisasi pada kasus *cybercrime*. Selain itu, identifikasi masalah yang ditemukan pada penelitian sebelumnya bahwa perlunya pengembangan lanjut terhadap sistem manajemen untuk memaksimalkan fungsi dan tujuan dari manajemen CoC. Upaya ini dilakukan untuk menemukan bagian utama sistem manakah yang dapat memberikan efektivitas dan efisien pada manajemen CoC. Sehingga dengan identifikasi masalah ini, perancangan konsep *on-chain* dan *off-chain* pada IPFSChain memang dibutuhkan. Adapun HF mendefiniskannya atas dasar permasalahan seperti komunikasi, identitas, konsensus, kontrak pintar atau *chaincode*, kriptografi, penyimpanan buku besar atau *ledger*, dan kebijakan (Casey et al., 2015). Di sisi lain, IPFS memiliki volume sistem penyimpanan yang besar dan *distributable*. Atas dasar dua sistem tersebut, yang memiliki tantangan tersendiri dan bersifat *opensource* serta dapat didistribusikan menjadi motivasi utama untuk melakukan penelitian ini.

3.3 Pengumpulan Data dan Studi Literatur

Pengumpulan informasi dan literatur terkait bagaimana CoC dibangun dan informasi apa saja yang seharusnya disimpan yang menjadi standar informasi dari CoC. Selain itu, studi terkait teknologi *blockchain*, dan sistem penyimpanan manakah yang lebih tepat untuk solusi dari masalah tersebut. Kemudian dilakukan pemetaan menjadi beberapa bagian penting untuk mencari poin utama dari masing-masing tahapan penelitian. Karena itu, konsep perancangan yang seperti apa yang dapat dibentuk dan bagaimana agar dapat diimplementasikan sesuai rancangan.

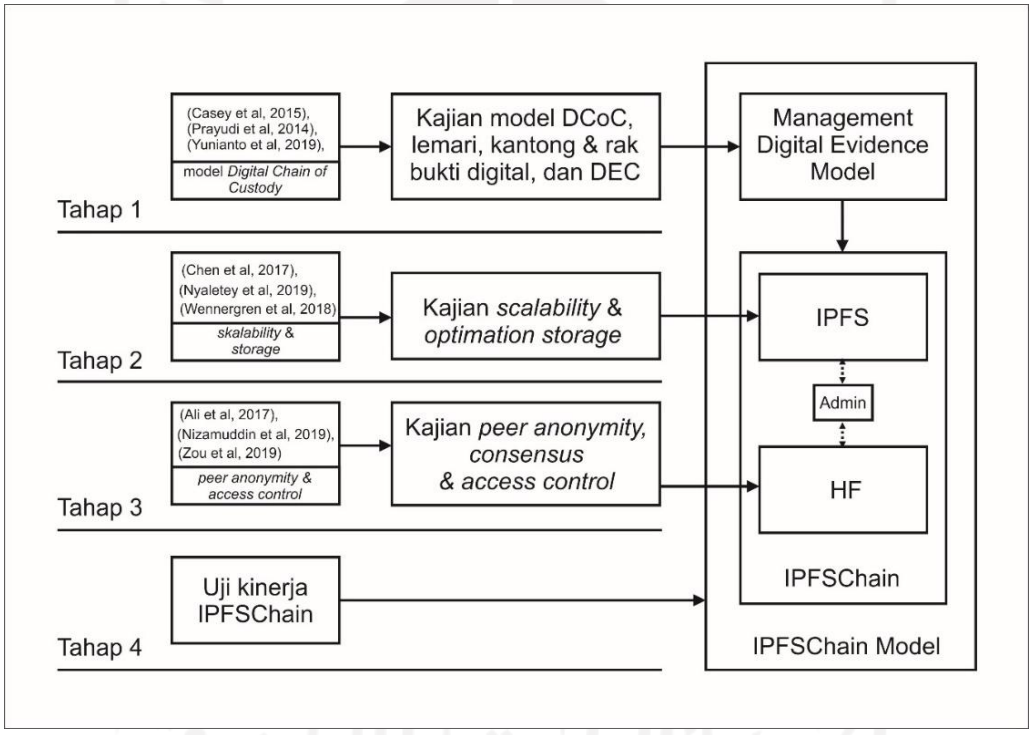
3.4 Desain Sistem IPFSChain

Bagian ini adalah analisis sistem, pembahasan tahap proses, dan desain sistem secara umum dan khusus terkait teknologi HF dan IPFS. Adapun istilah penggabungan kedua teknologi terdistribusi tersebut adalah IPFSChain model dengan konsep *off-chain* dan *on-chain*. Konsep tersebut mengacu pada penelitian yang dilakukan oleh (Nyaletey et al., 2019; Prayudi et al., 2014; Putra & Prayudi, 2021; Yunianto et al., 2019). Fokus utama pada bagian

ini adalah menyusun desain akses terhadap data yang lebih efektif, sehingga data hanya dapat terbaca dengan mudah oleh pihak-pihak yang telah ditentukan sebelumnya. Pendekatan yang dilakukan dengan *Digital Evidence Management Framework, distributed storage, access control, transparency,* dan keamanan transaksi.

3.4.1 Desain Umum

Perancangan model IPFSChain yang dibangun dengan konsep *off-chain* dan *on-chain. Off-chain* merupakan bagian yang bertujuan untuk tempat penyimpanan file asli dari bukti digital yaitu IPFS. Adapun *on-chain* adalah bagian penggunaan *blockchain* yang ditujukan sebagai keamanan pada informasi dari suatu bukti digital yang berupa *metadata*. Desain umum terdiri dari empat tahapan, selanjutnya informasi diberikan pada gambar 3.2.



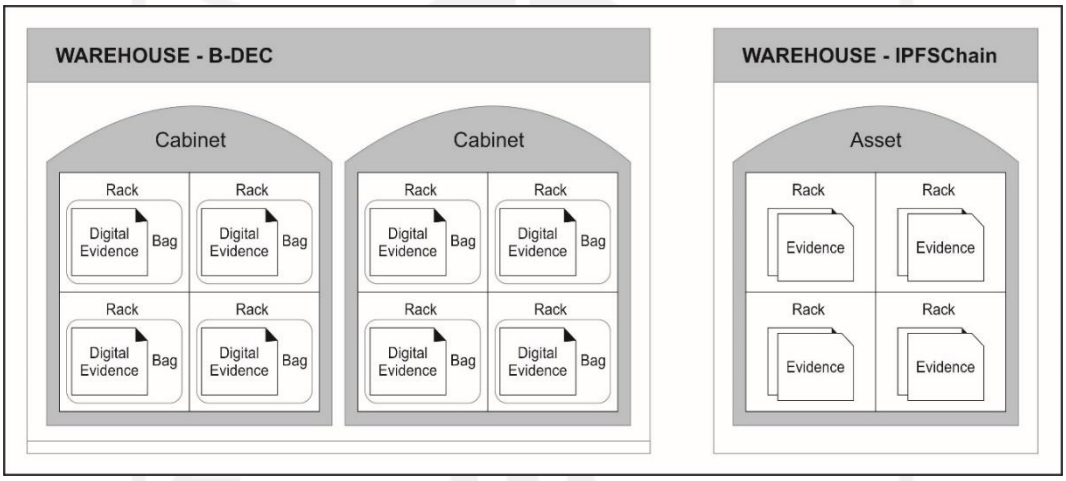
Gambar 3.2 Tahapan desain umum sistem IPFSChain.

A. Tahap I: Model Lemari, Rak dan DEC

Pembahasan terkait pendekatan dengan model manajemen bukti digital yang dalam hal ini adalah konsep DEC sebagai acuan dasar dalam mendesain konsep IPFSChain. Pada bagian ini akan berisi mengenai informasi apa saja yang perlu dicatat dari suatu bukti, dan model penyimpanan yang seperti apa yang dibutuhkan. Adapun model CoC pada DEC terdiri dari informasi metadata dinamis dan statis. Dari penelitian (Prayudi et al., 2014) dapat

disimpulkan, bahwa untuk informasi yang perlu dientri ke dalam dokumen CoC adalah informasi terkait nama kasus, nama dan format file, lokasi TKP, petugas TKP, pemilik bukti digital, proses akuisisi, jenis bukti digital, dan nilai *hash*. Informasi tersebut yang akan dientri ke dalam CoC berupa metadata.

Output pada tahapan ini adalah berupa kerangka lemari yang selanjutnya disebut sebagai aset. Aset atau lemari ini terdapat beberapa rak. Rak tersebut bertujuan untuk pengelompokan bukti digital berdasarkan jenis kasus. Dimana dalam rak terdapat satu atau banyak data yang direpresentasikan sebagai CoC. Dengan kata lain, informasi terkait satu bukti digital akan didokumentasikan menjadi satu CoC. Adapun perbandingan desain pendekatan lemari DEC oleh (Yunianto et al., 2019) dan IPFSChain terlihat pada gambar 3.3.



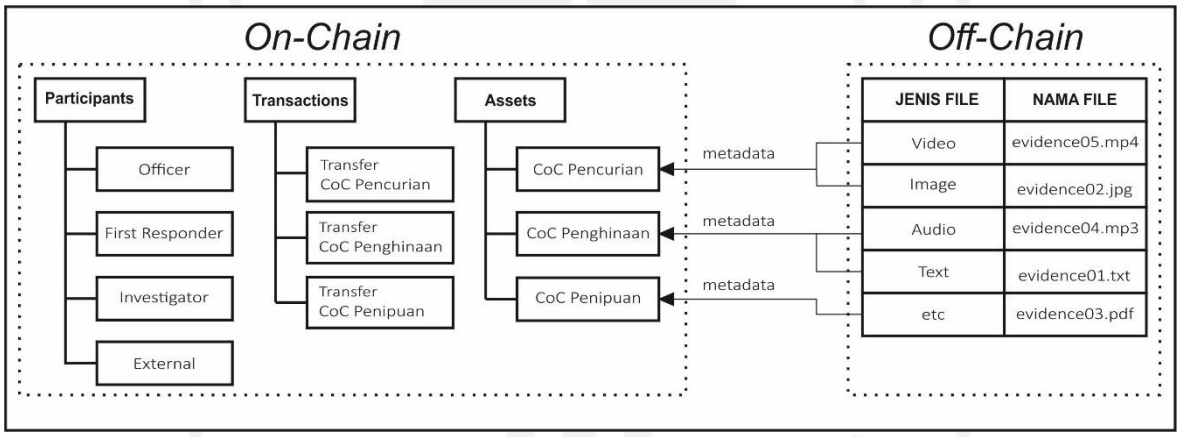
Gambar 3.3 Perbandingan desain lemari B-DEC dan IPFSChain.

B. Tahap II: IPFS sebagai Konsep *Off-Chain*

Bagian ini merupakan tahap penyimpanan file bukti digital yang dapat dilakukan oleh setiap peserta. Pada tahap ini, dimana kontrol akses tidak dapat diterapkan. Celah ini dialternatifkan ke *Hyperledger* sebagai solusi untuk memberikan keamanan pada file tersebut, yaitu berupa metadata file yang diikat ke dalam blok. Sedangkan file asli dari bukti digital disimpan ke dalam sistem IPFS. Gambar 3.4 menguraikan model alur interaksi pengguna dengan sistem IPFS.

Gambar 3.4 Interaksi pengguna dengan IPFS

File disimpan oleh *user* ke dalam IPFS untuk mendapatkan *link* akses berupa *multi-hash* berikut `Qmd4vF6R7GfKqhPVPdakL3cKD5YhNUrwdhTRc88UigQbxM`. *Multi-hash* tersebut adalah *link* untuk akses file evidence02.jpg sebagai representasi file bukti digital lainnya. Selanjutnya *multi-hash* ini disimpan dalam sistem HF. Dalam hal ini, IPFS adalah sebagai wadah penyimpanan data alternatif yang bersifat *off-chain* atau tidak terintegrasi dengan HF. Sistem HF dan IPFS tidak diintegrasikan karena sistem dari HF itu pada dasarnya adalah bersifat pencatatan yang permanen, maka dimungkinkan turunnya performa dari sistem HF itu sendiri seiring bertambahnya file/data yang disimpan dalam sistem IPFS. Dengan kata lain, segala aktivitas yang dilakukan dalam sistem IPFS akan tercatat oleh sistem HF. Maka sistem HF ini akan menyimpan pada *database/ledger* segala informasi yang terjadi pada IPFS. Sebagaimana yang dilakukan oleh (Nyaletey et al., 2019), bahwa rekam jejak pada IPFS akan tercatat oleh sistem HF.



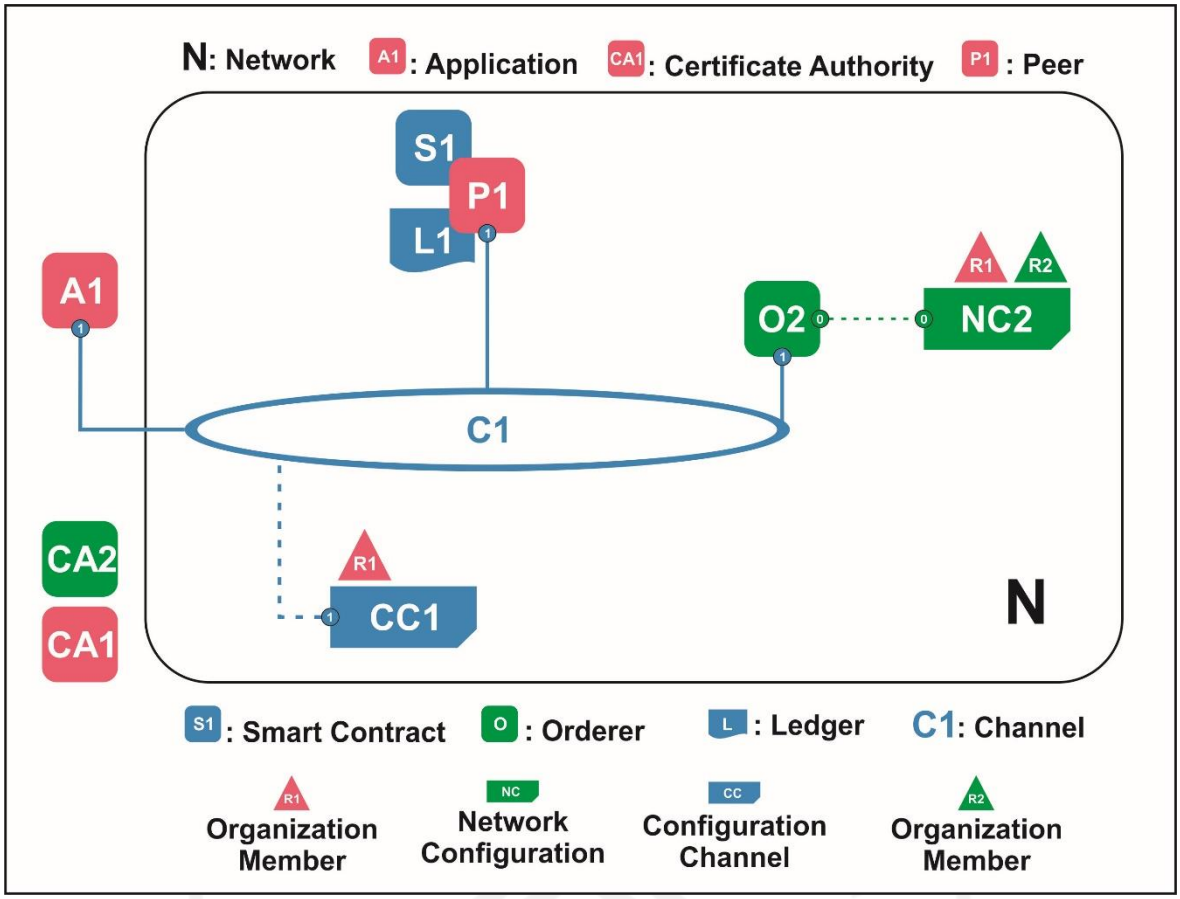
Gambar 3.5 Korelasi konsep *on-chain* dan *off-chain*

Gambar 3.5 menguraikan bahwa pada konsep *on-chain* terdiri dari beberapa komponen yang digunakan dalam perancangan jaringan bisnis dengan bantuan *tool hyperledger composer*. Dalam hal ini, IPFS sebagai konsep *off-chain* yang berkorelasi dengan konsep *on-chain* berupa metadata. Pada komponen aset, desain kabinet/aset terdiri dari tiga rak, yaitu rak CoC_Pencurian, CoC_Penghinaan, dan CoC_Penipuan. Sebagai representasi, rak CoC_Pencurian terdapat dua bukti digital yang berupa metadata. Sehingga jumlah aset yang disimpan pada tiga rak tersebut adalah lima aset.

C. Tahap III: HF Sebagai Kerangka Konsep *On-Chain*

Pada tingkat konseptual, tahap ini menjelaskan bagaimana HF memungkinkan organisasi untuk berkolaborasi dalam pembentukan jaringan *blockchain.* Adapun struktur utama dan komponen proses dalam jaringan *blockchain* HF ini ditunjukkan pada gambar 3.6.

1. Struktur dan Komponen *Hyperledger Fabric*



Gambar 3.6 Struktur dan komponen HF (hyperledger-fabric.readthedocs.io)

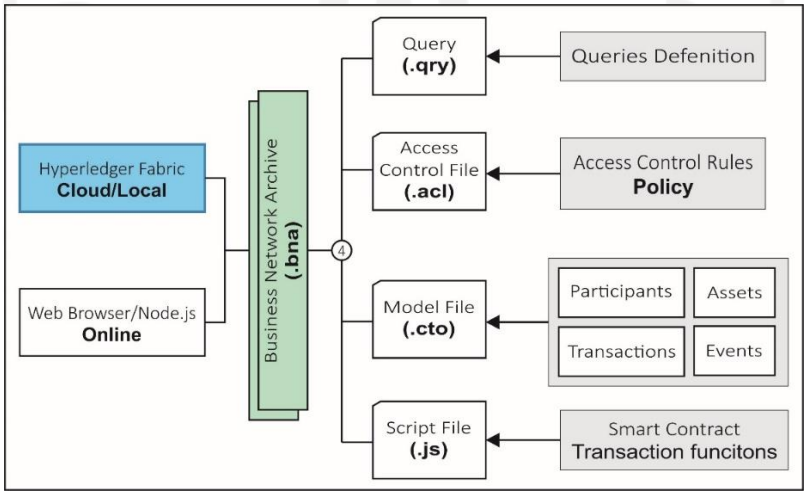
Jaringan terbentuk ketika *orderer* dimulai. Pada gambar 3.6 terlihat satu *node orderer* tunggal yang diberi nama O2, bahwa O2 dikonfigurasikan menurut NC2. NC2 berisi kebijakan konfigurasi awal yang memberikan hak administratif untuk jaringan atau dalam hal ini kepada organisasi R2. Pada tingkat jaringan, Otoritas Sertifikat CA2 digunakan untuk memberikan identitas kepada administrator dan *node* jaringan R2. Pemetaan sertifikat ke organisasi anggota dicapai melalui struktur yang disebut *Membership Services Provider* (MSP). Namun pada diagram ini, MSP tidak ditunjukkan karena akan mengacaukan pemahaman tetapi sangat penting untuk dijelaskan. Selanjutnya, NC2 dikonfigurasi menggunakan MSP ini untuk mengidentifikasi properti sertifikat yang dibagikan oleh CA2

yang mengaitkan pemegang sertifikat dengan organisasi R2. Kemudian NC2 ini menggunakan MSP ini dalam kebijakannya untuk memberikan aktor dari R2 hak tertentu atas sumber daya jaringan.

Dengan kata lain, R2 adalah super admin pertama yang pada fase berikutnya mengizinkan organisasi R1 untuk mengelola jaringan. Pada penelitian ini, organisasi R1 yang akan mengelola jaringan adalah partisipan *officer*. Sehingga pada titik ini, organisasi R1 memiliki hak atas konfigurasi jaringan. Selanjutnya *channel* diatur oleh CC1 yang terpisah dari konfigurasi jaringan atau NC2. Artinya CC1 dikelolah oleh R2 yang memiliki hak atas C1. Namun R2 tidak memiliki hak apapun di CC1.

P1 adalah node rekan yang berisi konfigurasi identitas X.509 yang dikeluarkan oleh CA1 yang mengaitkan P1 dengan organisasi R1. CC1 berisi salah satu kebijakan yang menentukan misalnya apakah P1 (organisasi R1) dapat membaca dan/atau menulis di saluran C1. Adapun komponen pada diagram diatas yang berwarna hijau adalah komponen super admin yang berdiri sendiri yang tidak memiliki node rekan. Selanjutnya, *smart contract* S1 atau *chaincode* diinstal ke P1. Aplikasi klien A1 dalam organisasi R1 dapat menggunakan S1 untuk mengakses data di *ledger* melalui node rekan P1. Pada diagram tersebut, A1 berada diluar jaringan *blockchain Fabric*. Namun demikian, A1 dapat terhubung ke node rekan P1 dan node pemesan O2 melalui saluran C1. S1 ini berfungsi sebagai pendefinisi semua pola akses umum ke buku besar L1. Dengan kata lain, A1 harus melalui S1 untuk sampai ke L1.

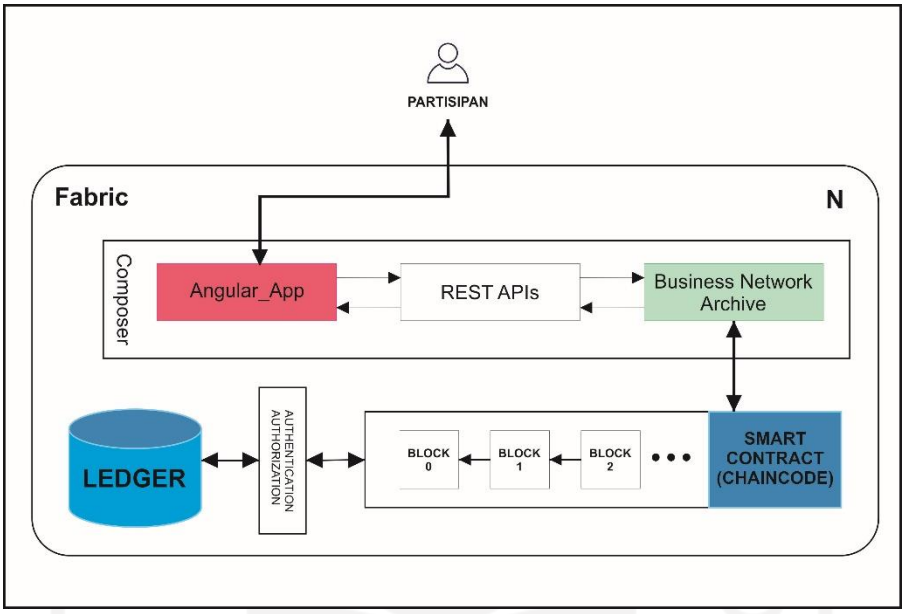
2. Struktur dan Komponen Hyperledger Composer



Gambar 3.7 Struktur dan komponen *Hyperledger Composer*

Gambar 3.7 menguraikan komponen utama pada *Hyperledger composer* adalah file **.bna**. *Business Network Archive* (BNA) ini adalah *smart contract* atau *chaincode* yang akan

dijalankan pada kerangka HF. Dalam *modular Hyperledger composer, chaincode* atau *smart contract* diprogram menggunakan bahasa javascript. Selanjutnya diuraikan pada gambar 3.8 bentuk interaksi partisipan ke jaringan HF melalui *Hyperledger composer* secara lokal.



Gambar 3.8 Interaksi partisipan terhadap jaringan HF melalui *composer*

Gambar 3.8 secara umum bahwa, partisipan untuk dapat sampai pada *ledger* melalui aplikasi *angular web app* dari *composer*. Aplikasi tersebut adalah sebagai simbol A1 pada diagram HF di atas. Adapun file **.bna** adalah sebagai *smart contract* dengan simbol S1 yang terpasang pada node rekan P1. Node rekan P1 ini akan digunakan oleh pihak *officer, first responder, investigator,* dan *external*. File BNA ini terdapat beberapa komponen yang terlihat pada gambar 3.7. Dalam *Hyperledger, chaincode* didefenisikan sebagai *multi smart contract.* Dimana *smart contract* dapat didesain lebih dari satu yang merupakan salah satu komponen dari *chaincode*. Namun pada penelitian ini, didesain cukup satu *smart contract* yang berfungsi sebagai transaksi atau transfer aset yang merubah kepemilikan dari suatu aset.

3. Desain Otoritas

Adapun kontrol akses yang akan dirancang dan diikat kepada peserta yang tergabung dalam jaringan terdiri dari otoritas partisipan terhadap partisipan lainnya, otoritas partisipan terhadap aset, dan otoritas partisipan terhadap transaksi. Sehingga desain otoritas ini menjadi

acuan untuk pengujian sistem secara fungsionalitas. Selanjutnya informasi ditunjukkan pada tabel 3.1, 3.2, dan 3.3.

Tabel 3.1 Desain otoritas partisipan terhadap partisipan, dan partisipan terhadap transaksi

No	Participant	Participant															
		Officer				First Responder				Investigator				Extern			
		C	*R*	*U*	*D*	*C*	*R*	*U*	*D*	*C*	*R*	*U*	*D*	*C*	*R*	*U*	*D*
1	Officer (Admin)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	First Responder		✓				✓				✓				✓		
3	Investigator		✓				✓				✓				✓		
4	External		✓				✓				✓				✓		

Gambar 3.2 adalah bentuk perancangan otoritas pada partisipan terhadap partisipan lainnya. Otoritas yang didesain berdasarkan fungsi dan tugas masing-masing pihak dalam jaringan HF. Adapun keterangan dari operasi *create* adalah menambahkan peserta baru ke dalam jaringan. *Read* adalah operasi yang berfungsi sebagai peserta dapat melihat aset, melihat peserta lainnya yang tergabung di jaringan serta dapat melihat semua aktivitas yang terjadi pada sistem. *Update* adalah untuk melakukan modifikasi terhadap informasi dari partisipan. *Delete* operasi untuk menghapus partisipan dalam jaringan.

Tabel 3.2 Desain otoritas partisipan terhadap aset

No	Participant	Asset											
		CoC_Pencurian				CoC_Penghinaan				CoC_Penipuan			
		C	*R*	*U*	*D*	*C*	*R*	*U*	*D*	*C*	*R*	*U*	*D*
1	Officer (Admin)		✓		✓		✓		✓		✓		✓
2	First Responder		✓	✓			✓	✓			✓	✓	
3	Investigator	✓	✓	✓		✓	✓	✓		✓	✓	✓	
4	External		✓				✓				✓		

Tabel 3.2 merupakan rancangan otoritas partisipan terhadap aset. Aset ini sebagai objek partisipan untuk dapat berinteraksi dan bertransaksi antar partisipan dalam jaringan. Adapun keterangan dari otoritas partisipan terhadap aset yaitu *create* adalah operasi yang berfungsi untuk membuat atau menambahkan aset ke dalam jaringan HF. Fungsi pada operasi lainnya telah dijelaskan diatas. Fungsi *create* diberikan pada partisipan *investigator* sebagai awal pembuat aset yang menentukan ID dari aset dan menentukan pada rak mana aset tersebut akan disimpan. Apakah aset tersebut disimpan pada rak CoC_Pencurian atau rak lainnya. Artinya *investigator* menentukan aset tersebut disimpan pada rak penyimpanan berdasarkan jenis kasus. Sehingga akan terdapat beragam jenis bukti digital yang disimpan dalam suatu rak penyimpanan. Sedangkan fungsi *update* diberikan pada *first responder*. Fungsi ini adalah untuk melengkapi informasi yang belum lengkap dari *investigator*.

Apabila informasi telah lengkap, *investigator* diberikan hak untuk melakukan transaksi ke pihak *external* atau jaksa yaitu berupa hak melakukan transfer aset sehingga status kepemilikan aset sebelumnya menjadi pemilik yang baru yaitu pihak jaksa. Adapun hak transfer tersebut berupa fungsi operasi *create* diuraikan pada tabel 3.3 dan fungsi operasi *update* yang diuraikan pada tabel 3.2. Dengan kata lain, aset ini menjadi catatan bersama. Dimana peserta yang diberikan hak tertentu dapat mengerjakan satu atau lebih aset secara bersamaan. Dari sisi ini dapat memberikan nilai transparansi yang jelas terhadap suatu tindakan yang dilakukan dalam jaringan.

Tabel 3.3 Desain otoritas partisipan terhadap transaksi

No	Participant	Transaction		
		CoC_Pencurian	**CoC_Penghinaan**	**CoC_Penipuan**
		Create	***Create***	***Create***
1	Officer (Admin)			
2	First Responder			
3	Investigator	✓	✓	✓
4	Extern			

Tabel 3.3 adalah otoritas partisipan terhadap transaksi. Transaksi ini merupakan salah satu fungsi *smart contract* yang didesain dalam *chaincode* yaitu operasi yang dapat mentransfer aset ke peserta lainnya dalam jaringan HF. Artinya apabila transaksi berhasil, maka status kepemilikan dari aset berubah dan valid bahwa pemilik aset sebelumnya telah memberikan hak kepemilikan asetnya ke partisipan yang di targetkan atau pemilik aset yang baru.

D. Tahapan Keempat: Pengujian Sistem IPFSChain

Pada tahap akhir ini, dilakukan serangkaian uji sistem dengan pendekatan *Black Box Testing* dan *White Box Testing.* Selanjutnya informasi ditunjukkan pada tabel 3.5.

Tabel 3.4 Data dan pengujian

Aktivitas	**IPFSChain**		**Pengujian**
	IPFS (file)	**Hyperledger (Aset)**	
Model dokumentasi, distribusi dan penyimpanan IPFSChain dengan konsep *on-chain* dan *off-chain*	evidence05.mp4 (50000KiB)	CoC_Pencurian	- White Box Testing - Black Box Testing
	evidence02.jpg (50KiB)		
	evidence01.txt (5KiB)	CoC_Penghinaan	
	evidence04.mp3 (5000KiB)		

	evidence03.pdf (500KiB)	CoC_Penipuan	

3.4.2 Analisis Sistem

A. Analisis Otoritas Pengguna Sistem

Otoritas ditentukan berdasarkan tugas masing-masing pengguna atau partisipan yang tergabung ke dalam sistem. Tujuan tersebut adalah untuk memberikan keamanan dan kontrol akses pada aset dalam sistem. Adapun partisipan atau pengguna yang akan didaftarkan ke dalam sistem terdiri dari empat pihak. Pada pihak investigator terdiri dari dua anggota. Ini dilakukan untuk mempermudah pemahaman terhadap otoritas yang diterapkan, dimana satu anggota dari pihak investigator dapat mengerjakan satu atau lebih bukti digital. Informasi tersebut diuraikan pada tabel 3.6.

Tabel 3.5 Anggota yang didaftarkan ke dalam sistem

No	Pihak / Partisipan	Anggota
1	*Officer*	Samri N
2	*First Responder*	Romi N
3	*Investigator*	Agus N
		Krisno N
4	*External*	Nurdin N

B. Analisis Kebutuhan *Input* Sistem

Kebutuhan input sistem yaitu:

1. Input identitas partisipan pada sistem HF berdasarkan peran dan tugas yang telah ditentukan.
2. Input metadata bukti digital ke dalam rak CoC berdasarkan kasus dalam sistem HF.
3. Input file bukti digital pada sistem IPFS.
4. Input *multi-hash* IPFS berupa *link* akses ke sistem HF.

C. Analisis Kebutuhan Proses

Proses yang akan dilakukan untuk kemudahan akses, dan transfer serta bersifat transparansi yaitu:

1. Proses *create* atau pembuatan ID *Card* partisipan.
2. Proses *create* atau entri informasi bukti digital pada rak CoC berdasarkan kasus tertentu.
3. Proses *add* file bukti ke dalam sistem IPFS dan secara otomatis IPFS memberikan *link* akses dalam bentuk *multi-hash.*

4. Proses entry *multi-hash* dari IPFS ke dalam sistem HF.
5. Proses transaksi berupa transfer aset dalam sistem HF.
6. Proses *get* file bukti dari IPFS.
7. Proses *update* dan *delete* aset pada HF dan proses *delete* bukti digital pada IPFS.
8. Proses penyimpanan *log* akses terhadap aset secara otomatis dalam HF.
9. Proses menampilkan *log* aktivitas dalam HF secara transparan.

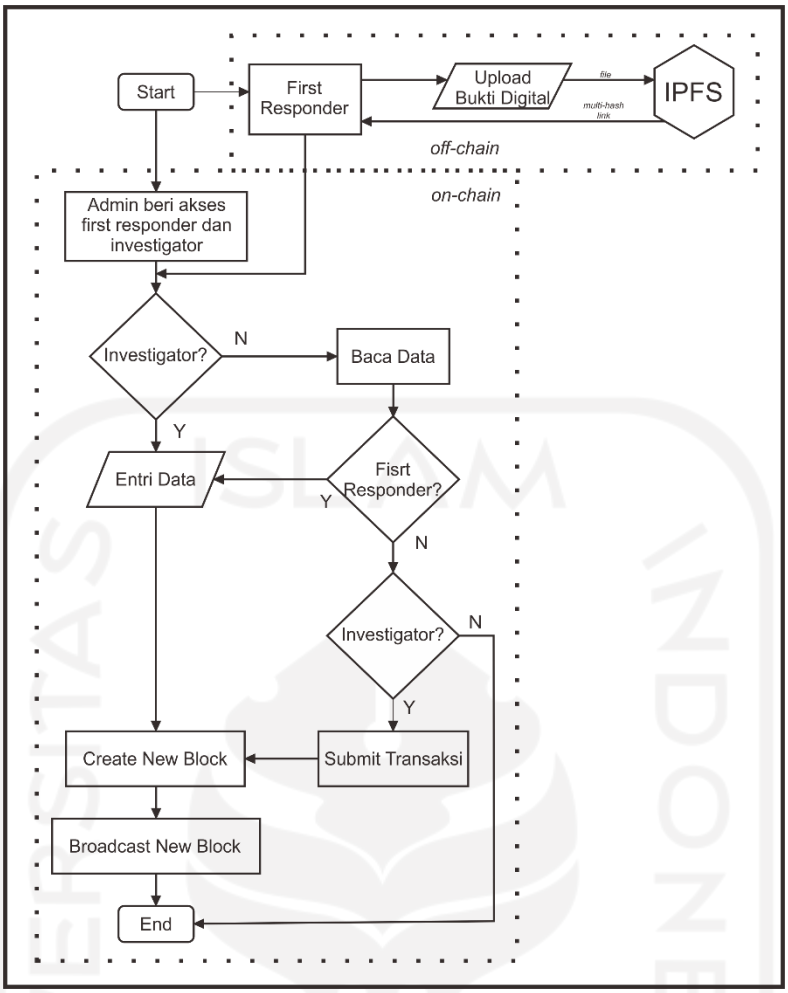
D. Analisis Kebutuhan *Output* Sistem

Dari rangkaian proses penggunaan sistem IPFSChain dalam mengelola bukti digital dan CoC dihasilkan beberapa *output* yaitu:

1. File bukti digital dapat diakses oleh partisipan ataupun siapapun yang mendapatkan link akses dari IPFS berupa *multi-hash.*
2. *Log* aktivitas bersifat transparan dapat terlihat oleh semua partisipan yang tergabung dalam jaringan HF.

3.4.3 Diagram Alir IPFSChain

Berikut adalah gambaran alur sistem yang akan dirancang dan diimplementasikan dengan konsep *on-chain* dan *off-chain*. Otoritas ditentukan berdasarkan tugas masing-masing pengguna atau partisipan yang tergabung ke dalam sistem.

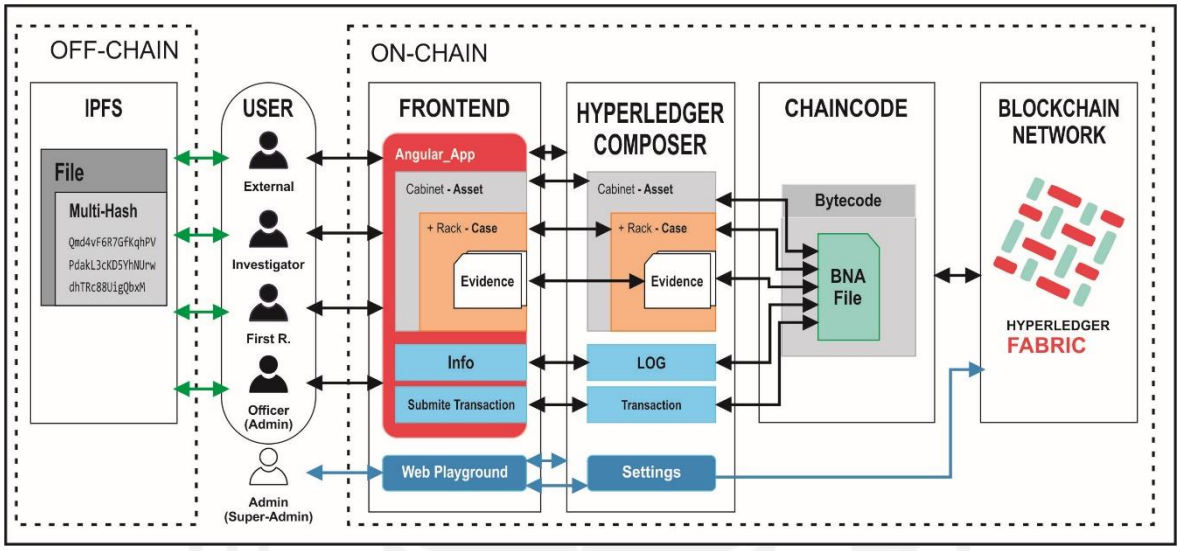


Gambar 3.9 Diagram alir penyimpanan bukti digital dan pendokumentasian CoC pada IPFSChain model

Proses penyimpanan dimulai dari *first responder* melakukan *upload* file bukti digital ke sistem IPFS untuk kemudahan akses file tersebut oleh partisipan lain. Selain itu, sistem IPFS secara otomatis akan mengeluarkan *link* akses berupa *multi-hash*. Selanjutnya proses pencatatan dan pendokumentasian, admin memberikan akses kepada pihak *investigator* dan *first responder*. Apabila pihak yang terlibat adalah *investigator*, maka pihak tersebut melakukan entri ID aset. Jika tidak, maka hanya pihak *first responder* yang diberikan otoritas untuk pemutakhiran aset yang belum lengkap. Apabila pihak yang terlibat bukan dari dua pihak yang disebutkan, maka pihak tersebut hanya dapat membaca. Adapun pihak *officer* hanya dapat membaca dan menghapus aset. Status blok akan selalu termutakhirkan ketika beberapa operasi *create/write, update, delete,* dan submit transaksi berhasil dijalankan.

3.4.4 Desain Arsitektur IPFSChain

IPFSChain model dibangun atas dasar kemudahan penyimpanan, akses dan transfer data dengan memprioritaskan transparan, kontrol akses dan keamanan, serta ketertelusuran pada semua aktivitas dalam jaringan. Maka IPFSChain model ini didesain dalam bentuk *on-chain* dan *off-chain*.



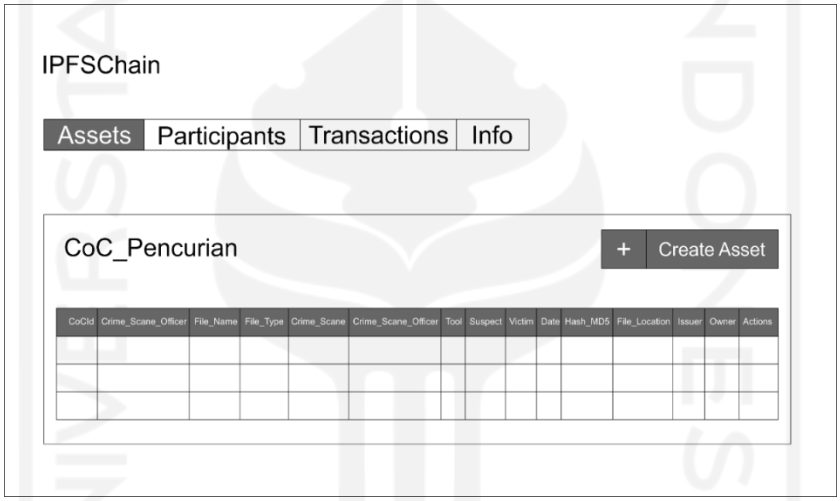
Gambar 3.10 Arsitektur IPFSChain model

Arsitektur IPFSChain model pada gambar 3.10 menguraikan bahwa, *user* berinteraksi dengan sistem IPFS dengan dua cara, yaitu *upload* dan *download. User* adalah persona yang belum terdaftar di jaringan HF atau calon partisipan. Partisipan adalah persona yang telah terdaftar di jaringan HF. Pada penelitian ini, pihak *first responder* yang bertugas untuk *upload* file bukti digital ke IPFS. Secara otomatis IPFS mengenerate file yang diunggah menjadi *multi-hash* sebagai *link* akses. *Multi-hash* ini yang akan disimpan atau dituliskan oleh *first responder* ke dalam jaringan HF melalui *Hyperledger composer*. *Multi-hash* tersebut adalah salah satu bagian informasi yang dituliskan dalam aset CoC. Secara umum aset adalah objek yang bergerak di antara partisipan atau dapat direpresentasikan dalam bentuk biner dan/atau JSON . Sedangkan aset berdasarkan gambar 3.5 adalah sebagai wadah atau lemari yang terdapat beberapa rak untuk menyimpan *evidence* berupa metadata. Selanjutnya metadata ini dalam sistem *Hyperledger* disebut aset. Adapun bentuk kerja seorang partisipan dengan aset adalah dengan melakukan transaksi. Transaksi adalah fungsi jaringan dan dipanggil untuk memperbarui jaringan. Adapun transaksi pada IPFSChain adalah upaya antara pihak *first responder*, *officer*, *investigator*, dan *external* dalam mengirimkan aset, sehingga secara khusus status aset akan berubah kepemilikan dan status

jaringan pada umumnya otomatis akan termutakhirkan. Namun pada penelitian ini, hanya pihak *investigator* yang diberikan otoritas untuk melakukan transaksi yaitu berupa transfer aset.

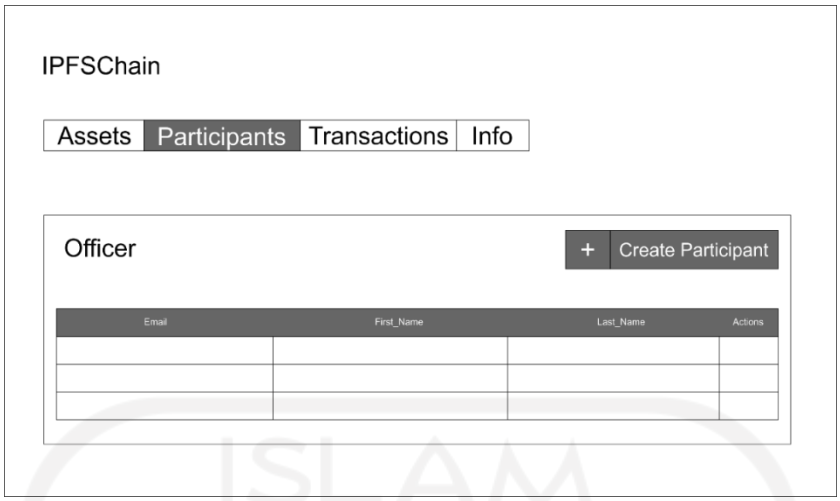
3.4.5 Desain Antarmuka Sistem

Pada bagian ini, desain antarmuka sistem IPFSChain model akan dibangun dengan pendekatan sistem pada sisi klien berbasis *web* menggunakan *tool hyperledger composer* dengan *generators* RESTAPIs dan *angular web app*. Adapun pendekatan sistem pada sisi admin menggunakan *web playground* untuk simulasi kasus. Berikut adalah desain antarmuka pengguna dengan *angular web app generator.*



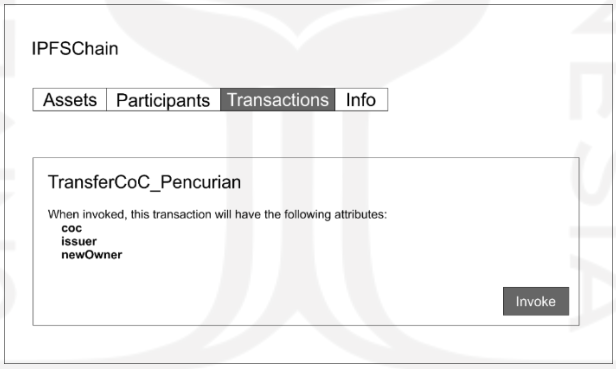
Gambar 3.11 Desain halaman *create asset* pada rak CoC_Pencurian

Melalui antarmuka diatas sebagai representasi rak lainnya, rak CoC_Pencurian berisi data-data terkait kasus pencurian atau dengan kata lain adalah sebagai bentuk dokumen digital CoC. Partisipan *investigator* yang memiliki akses menambahkan ID CoC dengan entry "coc01" dan entry metadata pada *form issuer* yang sesuai dengan email *investigator* yang terdaftar yaitu "ir.agus@mail.com". Selanjutnya partisipan *first responder* dapat entry metadata dari suatu bukti digital pada form CoC01 yang masih kosong atau belum lengkap dan entry pada *form owner* dengan email *first responder* yang telah terdaftar yaitu fr.rom@mail.com.



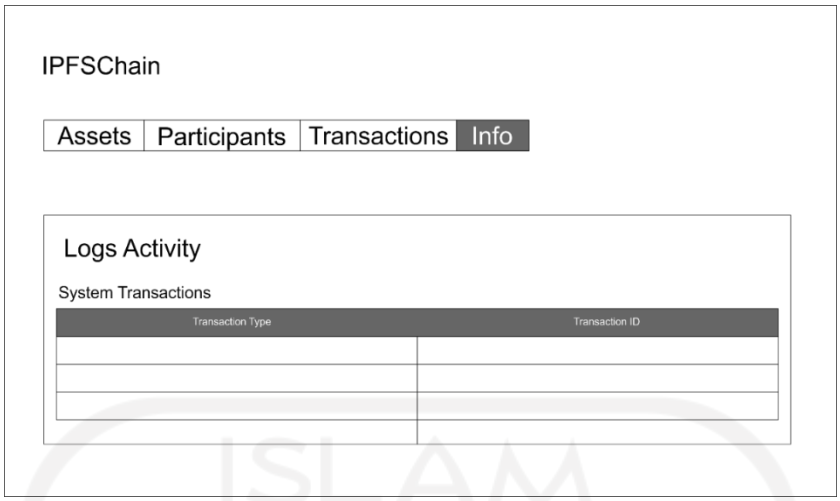
Gambar 3.12 Desain halaman *create participant*

Pada antarmuka diatas sebagai representasi partisipan lainnya, partisipan *officer* yang memiliki akses dan hak otoritas dapat menambahkan dan menghapus partisipan lainnya. Adapun partisipan yang memiliki akses dan tidak ada hak otoritas penuh atas partisipan lainnya, maka hanya dapat melihat.



Gambar 3.13 Desain halaman *transactions*

Transaksi hanya dapat dilakukan oleh pihak investigator. Transaksi yang dimaksud adalah berupa aktivitas transfer aset kepada pihak *external.* Tujuan akhir dari kepemilikan dari suatu aset adalah kepada pihak *external*. Apabila transaksi berhasil dilakukan, maka kepemilikan awal dari suatu aset yaitu pihak *investigator* sebagai *issuer* ke pihak *external* sebagai *newOwner.*

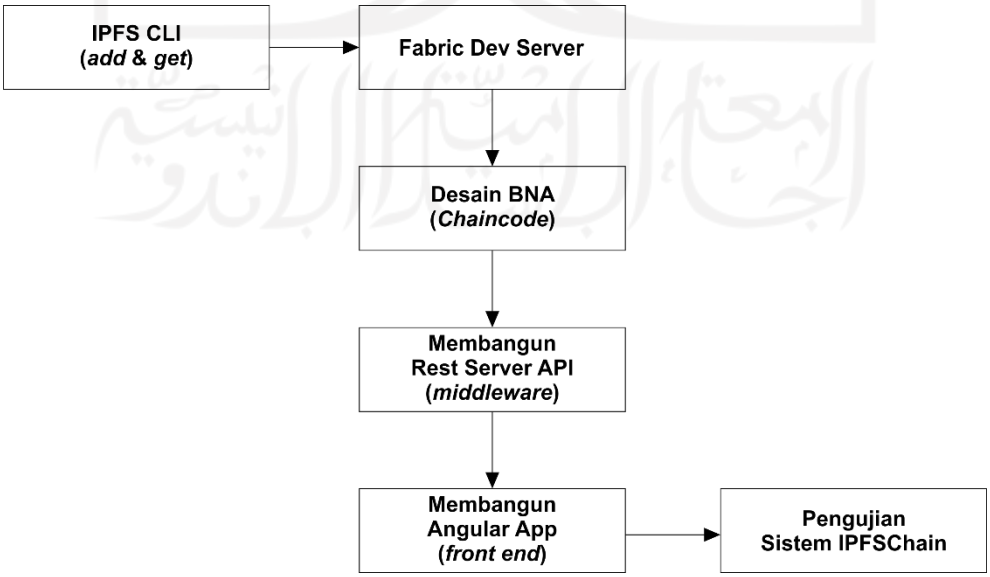


Gambar 3.14 Desain halaman *log activity*

Halaman *logs activity* pada menu info adalah untuk menampilkan semua aktivitas/interaksi partisipan dalam sistem secara transparan.

3.5 Implementasi Sistem

Tahapan dalam membangun sistem IPFSChain model untuk akses dan transfer data dalam jaringan antara lain dimulai dari sisi *off-chain* yaitu IPFS CLI yang terdapat *operation functions add*, dan *get*. Adapun tahapan pada sisi *on-chain* yang akan membangun material atau komponen melalui *fabric-dev-server*, desain *business network archive* (BNA) atau *smart contract*, membangun *rest_server* atau *middleware*, membangun *angular web app* atau *front end*. Selanjutnya pada tahap akhir adalah pengujian sistem.



Gambar 3.15 Alur implementasi IPFSChain model

3.5.1 IPFS CLI

IPFS CLI merupakan salah satu cara untuk berinteraksi dengan sistem IPFS melalui *command-line.* Bentuk perintah yang akan dilakukan pada bagian ini adalah *add,* dan *get.* Perintah *add* adalah fungsi operasi untuk menambahkan atau melakukan *upload* file ke IPFS. Berikut perintah *add* melalui *command-line*:

```
$ ipfs add evidence02.jpg
```

Selanjutnya, sistem IPFS memberikan nilai unik *multi-hash* Qmd4vF6R7GfKqhPVPdakL3cKD5YhNUrwdhTRc88UigQbxM dari file evidence02.jpg sebagai link untuk akses file. Nilai unik *multi-hash* tersebut yang akan disimpan ke HF untuk diikat sebagai alamat lokasi dimana file tersebut disimpan. Perintah untuk akses/unduh file tersebut menggunakan *get*. Berikut perintah get melalui *command-line*:

```
$ ipfs get /ipfs/Qmd4vF6R7GfKqhPVPdakL3cKD5YhNUrwdhTRc88UigQbxM
```

Adapun akses/unduh file juga dapat dilakukan melalui browser dengan menempelkan url ipfs.io/ipfs/Qmd4vF6R7GfKqhPVPdakL3cKD5YhNUrwdhTRc88UigQbxM.

3.5.2 *Fabric Dev Server*

Fabric dev server terdapat beberapa komponen yang akan menjalankan fungsinya untuk dapat berkomunikasi antara *platform fabric* dengan *tool composer.* Ini bertujuan untuk menjalankan/menyebarkan jaringan bisnis secara lokal. Adapun komponen utama yang dibutuhkan pada tahap ini adalah *composer-cli, composer-rest-server, generator-hyperledger-composer, yeoman (yo), composer-playground,* dan *fabric-dev-server* itu sendiri. Adapun komponen yang terdapat dalam *fabric-dev-server* adalah *downloadFabric.sh, teardownFabric.sh, startFabric.sh, stopFabric.sh, createPeerAdminCard.sh.* Komponen ini sebagai material *fabric* untuk dapat berkomunikasi dengan *tool hyperledger composer.*

3.5.3 Desain *Business Network Archive* (BNA)

Business network archive adalah sebagai logika bisnis atau *smart-contract* yang akan dijalankan pada jaringan *fabric.* BNA adalah format dari suatu file yaitu .bna yang terdiri dari kumpulan file akses kontrol (*permission.acl*), model (*model.cto*), dan *java script*

(*logic.js*). Terlihat pada gambar 3.7 dan 3.8. Otoritas pada tabel 3.1, 3.2, dan 3.3 didesain dan disimpan dalam file *permission.acl.* File *model.cto* berisi informasi metadata yang didesain terkait partisipan, aset, dan transaksi. Adapun file *logic.js* berisi logika bisnis sebagai *smart contract.* Tiga file tersebut saling terhubung satu dengan lainnya yang selanjutnya dalam *hyperledger* disebut dengan *chaincode.*

3.5.4 Membangun *Rest Server API*

Server REST hyperledger composer digunakan untuk menghasilkan *REST API* dari jaringan. Aplikasi web harus melakukan panggilan REST API untuk dapat berinteraksi dengan jaringan bisnis yang akan digunakan. Adapun perintah yang digunakan untuk membuat REST API yaitu dengan mengetikan *composer-rest-server* pada *command line.* Selanjutnya, entri nama kartu jaringan untuk admi. *Generated API namespace, secure generated API, enable event publication, enable TLS security,* dan *localhost* yang digunakan.

3.5.5 Membangun *Front-end*

Tahap ini adalah membangun antarmuka dengan aplikasi *angular*. Aplikasi *angular* memerlukan server lain untuk dapat terhubung ke *fabric* yaitu dengan memastikan *REST server* aktif. Untuk dapat melakukan *generate* dan menjalankan aplikasi *angular* dengan menjalankan generator *yeoman* dan harus berada dalam direktori yang sama dengan file .bna.

3.6 Pengujian Sistem

Pengujian sistem yang diuji dengan beberapa jenis data dengan ukuran yang berbeda dan sejumlah model pengujian tersendiri. Adapun pengujian pada sistem IPFS berfokus pada aktivitas *add*, dan *get*. Sedangkan pada sistem HF berfokus pada *create, read, update,* dan *delete.* Tabel 3.3 menunjukkan data dan pengujian pada tahapan penelitian yang akan dilakukan.

3.6.1 Pengujian Fungsionalitas

Tahapan ini, pengujian sistem didasarkan pada fungsionalitas sistem yang diakomodasi oleh IPFSChain. Apakah sistem pendokumentasian dan pendistribusian terhadap suatu bukti digital berjalan sesuai fungsi atau tidak. Sebelum itu, dibutuhkan desain otoritas atau kontrol akses partisipan terhadap partisipan, partisipan terhadap aset, dan partisipan terhadap transaksi. Informasi ditunjukkan pada tabel 3.4 dan 3.5. Adapun tabel 3.6, menunjukkan

skenario pengujian fungsionalitas yang akan dilakukan pada sistem HF berdasarkan otoritas pada tabel 3.4 dan 3.5.

Tabel 3.6 Daftar skenario pengujian fungsionalitas

No	Partisipan	Aktivitas	Izin
1	Officer (Admin)	Melakukan *Create, Read, Update, Delete* pada *Officer*	Bisa
		Melakukan *Create, Read, Update, Delete* pada *First Responder*	Bisa
		Melakukan *Create, Read, Update, Delete* pada *Investigator*	Bisa
		Melakukan *Create, Read, Update, Delete* pada *External*	Bisa
		Melakukan *Read, Delete* pada aset CoC_Pencurian	Bisa
		Melakukan *Read, Delete* pada aset CoC_Penghinaan	Bisa
		Melakukan *Read, Delete* pada aset CoC_Penipuan	Bisa
		Melakukan *Create* pada *Transfer* CoC_Pencurian	Tidak Bisa
2	First Responder	Melakukan *Read* pada *Officer*	Bisa
		Melakukan *Read* pada *First Responder*	Bisa
		Melakukan *Read* pada *Investigator*	Bisa
		Melakukan *Read* pada *External*	Bisa
		Melakukan *Read, Update* pada Aset CoC_Pencurian	Bisa
		Melakukan *Read, Update* pada Aset CoC_Penghinaan	Bisa
		Melakukan *Read, Update* pada Aset CoC_Penipuan	Bisa
		Melakukan *Create* pada *Transfer* CoC_Pencurian	Tidak Bisa
3	Investigator	Melakukan *Read* pada *Officer*	Bisa
		Melakukan *Read* pada *First Responder*	Bisa
		Melakukan *Read* pada *Investigator*	Bisa
		Melakukan *Read* pada *External*	Bisa
		Melakukan *Create, Read, Update* pada Aset CoC_Pencurian	Bisa
		Melakukan *Create, Read, Update* pada Aset CoC_Penghinaan	Bisa
		Melakukan *Create, Read, Update* pada Aset CoC_Penipuan	Bisa
		Melakukan *Create* pada Transfer CoC_Pencurian	Bisa
4	External	Melakukan *Read* pada *Officer*	Bisa
		Melakukan *Read* pada *First Responder*	Bisa
		Melakukan *Read* pada *Investigator*	Bisa
		Melakukan *Read* pada *External*	Bisa
		Melakukan *Read* pada Aset CoC_Pencurian	Bisa
		Melakukan *Read* pada Aset CoC_Penghinaan	Bisa
		Melakukan *Read* pada Aset CoC_Penipuan	Bisa
		Melakukan *Create* pada *Transfer* CoC_Pencurian	Tidak Bisa

3.6.2 Pengujian Kinerja IPFSChain

Tahapan ini, pengujian sistem didasarkan pada efektivitas atau kemudahan akses dan transfer data. Pengamatan pada beberapa proses eksperimen terhadap sistem memiliki sejumlah model pengujian tersendiri. Pengujian aktivitas *create* dan/atau *delete* pada tiap komponen HF ditentukan dari skema uji yang dibutuhkan. Tabel 3.7 menguraikan parameter ukur dan nilai dari setiap ketentuan skema uji. Adapun pengujian aktivitas *add* dan *get* pada sistem IPFS menggunakan beberapa jenis data dan ukuran yang berbeda. Informasi tersebut terlihat pada tabel 3.8.

Tabel 3.7 Pengujian kinerja HF

No	Components	Activities	Size (KB)	Avg. Latency (s)
1	Partisipan	Tambah 1 anggota *officer*		
		Tambah 1 anggota *first responder*		
		Tambah 2 anggota *investigator*		

		Tambah 1 anggota *external*		
		Hapus 1 anggota dari salah satu pihak		
2	Aset CoC	Tambah 2 aset ke dalam rak CoC_Pencurian		
		Tambah 2 aset ke dalam rak CoC_Penghinaan		
		Tambah 1 aset ke dalam rak CoC_Penipuan		
		Hapus 1 aset dari salah satu rak		
3	Transfer CoC	Submit transaksi 1 aset		

Tabel 3.8 Pengujian kinerja IPFS

No	Digital Evidence	Response Time (s)		Size (KB)
		Add	Get	
1	evidence01.txt			
2	evidence02.jpg			
3	evidence03.pdf			
4	evidence04.mp3			
5	evidence05.mp4			

Tabel 3.9 Hasil pengujian IPFSChain

No	Klausal	B-DEC	Multi Smart Contract	IPFSChain	Keterangan
1	*Authentication* pada halaman *login*				*Input username* dan *password* saat *login*
2	Mampu mengakomodir lebih dari satu bukti digital				Dalam satu kasus memiliki banyak bukti digital
3	Informasi bukti digital bersifat dinamis				Informasi dientri susai dengan kaidah CoC atau bersifat dinamis
4	*Smart contract* dinamis sesuai kebutuhan				Menyimpan metadata ke blok berdasarkan kasus
5	Bukti digital disimpan dalam sistem IPFS yang memiliki varian jenis dan ukuran				File asli hanya dapat diakses menggunakan *multi-hash* dari IPFS
6	Memiliki *log* aktivitas terhadap aset				Aksi input, update, akses, dan delete disimpan dalam blok
7	Partisipan tersertifikasi sebagai *signature* secara digital				*Card* ID berupa *key private* dan *certification authority* dari sistem HF

BAB 4
Hasil dan Pembahasan

4.1 Implementasi

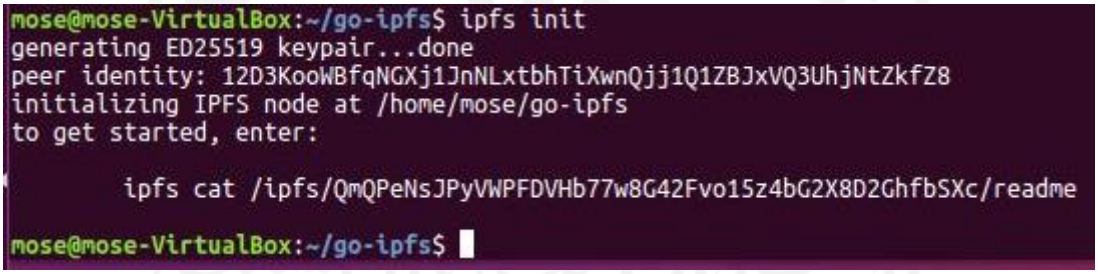
Kebutuhan utama sebelum implementasi konsep IPFSChain dimulai dari sistem operasi yang digunakan adalah ubuntu 16.04 LTS yang dijalankan dengan menggunakan VirtualBox versi 6.1, kapasitas penyimpanan 20GB, RAM 4GB, prosesor 1 *core*, koneksi jaringan NAT.

4.1.1 Membangun Konsep *Off-Chain*

Membangun konsep ini dimulai dari mengunduh dan install IPFS pada ubuntu 16.04 yang dijalankan dalam *virtual machine.* Berikut perintah yang dijalankan melalui *command line* untuk unduh dan install IPFS:

```
$ wget https://dist.ipfs.io/go-ipfs/v0.11.0/go-ipfs_v0.11.0_linux-amd64.tar.gz
$ tar -xvzf go-ipfs_v0.11.0_linux-amd64.tar.gz
$ cd go-ipfs
$ sudo bash install.sh
```

Setelah berhasil diinstal, sebelum melakukan unggah file bukti digital ke IPFS secara lokal terlebih dulu melakukan *generated peer ID* dengan menjalankan perintah `ipfs init` seperti berikut:



Gambar 4.1 Inisialisasi repositori

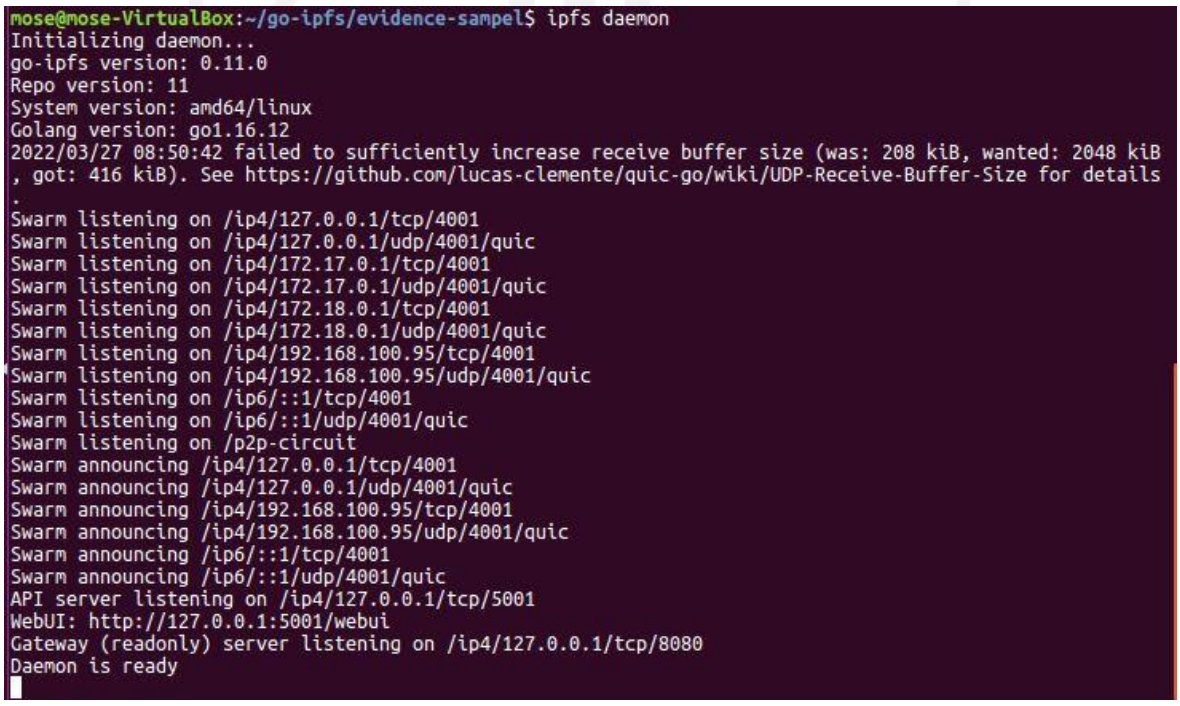
IPFS menyimpan semua pengaturan dan data internalnya dalam direktori yang disebut repositori. Masih dalam direktori yang sama, yaitu go-ipfs. Dalam direktori ini dibuat sebuah direktori baru yang diberi nama evidence-sample untuk menyimpan lima jenis bukti digital yang akan digunakan dan disimpan ke IPFS. Selanjutnya, masuk ke folder *evidence-sample.* Berikut adalah perintah yang akan dijalankan untuk melakukan *add,* dan *get* pada file evidence02.jpg sebagai representasi file lainnya.

Gambar 4.2 Unggah file bukti digital ke IPFS

Pada gambar 4.2 terlihat bahwa sistem IPFS secara otomatis memberikan nilai unik berupa *multi-hash* untuk file evidence02.jpg. *Multi-hash* ini sebagai alamat lokasi file evidence02.jpg disimpan. Selain itu juga sebagai link yang digunakan untuk akses/unduh file tersebut. Berikut perintah untuk unduh file evidence02.jpg.

Gambar 4.3 Unduh file bukti digital dari IPFS

Perintah di atas masih bersifat penyimpanan secara lokal. Untuk dapat file tersebut diakses secara global, maka perlu dijalankan perintah berikut:



Gambar 4.4 Menggabungkan *node* ke jaringan publik

4.1.2 Membangun Konsep *On-Chain*

Konsep ini dibangun dengan jaringan *fabric* berupa *single organization* melalui modular *hyperledger composer.* Tahap ini akan menguraikan langkah-langkah yang perlu diambil oleh administrator untuk menyebarkan jaringan bisnis *blockchain* ke *platform hyperledger fabric.* Adapun prasyarat yang diperlukan diuraikan pada tabel 4.1.

Tabel 4.1 Prasyarat lingkungan pengembangan

No	Komponen
1	Docker Engine v20.10
2	Docker-Compose v1.28
3	Node v8.15
4	Npm v6.4
5	Python v2.7
6	VSCode
7	CLI Tools v0.20
8	Composer-Playground v0.20
9	Yeoman v3.1.1
10	Fabric-dev-servers v1.2

A. Menjalankan *Hyperledger Fabric*

Untuk memulai jaringan *fabric*, berikut perintah dan keterangannya:

```
$ cd ~/fabric-dev-servers
Export FABRIC_VERSION=hlfv12
```

Perintah di atas adalah untuk menentukan tujuan komponen yang dioperasikan akan menuju ke jalur jaringan *hyperledger fabric* versi 1.2. Versi ini dianggap lebih stabil dibandingkan versi 1.1 atau versi 1

```
$ ./downloadFabric.sh
```

Perintah di atas adalah untuk mengunduh kebutuhan lingkungan *fabric* yaitu *hyperledger/fabric-peer:1.2.1, hyperledger/fabric-ca:1.2.1, hyperledger/fabric-ordere:1.2.1,* dan *hyperledger/fabric-couchdb:0.4.10* dan disimpan ke dalam kontainer *docker*.

```
$ ./startFabric.sh
$ ./createPeerAdmin.sh
```

Perintah di atas apabila berhasil dilakukan, maka jaringan *fabric* siap untuk dioperasikan sebagai *platform* untuk bisnis jaringan yang akan dibangun. Perintah `./createPeerAdmin.sh` adalah perintah membuat kartu jaringan bisnis *peer* admin untuk menyebarkan jaringan bisnis *hyperledger composer* ke *hyperledger fabric*.

B. Buat Struktur Jaringan Bisnis

Konsep utama yang menjadi kunci untuk *hyperledger composer* adalah *business network defenition* (BND). Ini mendefenisikan atau perancangan model data (*model.cto*), logika transaksi (*logic.js*), dan aturan kontrol akses (*permission.acl*). Adapun tahapannya sebagai berikut:

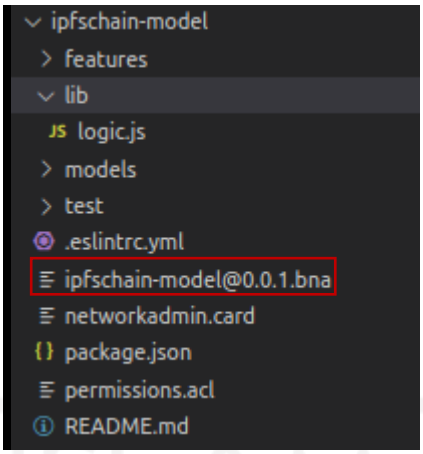
1. Membuat kerangka jaringan bisnis dengan menggunakan *yeoman. Command line* berikut: $ yo hyperledger-composer:businessnetwork. Perintah ini akan memerlukan nama jaringan bisnis, deskripsi, nama penulis, email penulis, pemilihan lisensi, dan *namespace.*
2. Nama jaringan bisnis: *ipfschain-model.*
3. Lisensi: *Apache-2.0*
4. *Namespace*: *org.example.empty*

Selanjutnya, pada tahap ini akan dilakukan perancangan dan konfigurasi dari tiga file di atas yang didefeniskan dibawah ini.

1. Model data (*model.cto*): mendefenisikan kelas (*class*) untuk semua aset, partisipan, dan transaksi. Terlampir pada halaman lampiran A.
2. Logika transaksi (*logic.js*): berisi fungsi prosesor transaksi. Terlampir pada halaman lampiran A.
3. Kontrol Akses (*permission.acl*): berisi dokumen kontrol akses dasar. Terlampir pada halaman lampiran A.
4. Metadata jaringan bisnis (*package.json*): berisi informasi terkait jaringan bisnis berupa metadata.

4.1.3 Menyebarkan BNA (*chaincode*)

Setelah jaringan bisnis ditentukan, kemudian dikemas ke dalam file *business network archive* (.bna) dengan perintah `composer archive create -t dir -n .` yang akan digunakan pada *platfotm fabric.* Kemudian akan mengahsilkan file ipfschain-`model`@0.0.1.bna. Terlihat pada gambar 4.5.



Gambar 4.5 File *business network archive*

Menyebarkan jaringan bisnis ke *fabric hayperledger* memerlukan arsif jaringan bisnis *hyperledger composer* atau file .bna tersebut untuk diinstall pada rekan. Berikut tahapanya:

1. Install file .bna dari direktori ipfschain-`model`.

```
composer network install –card PeerAdmin@hlfv1 –archiveFile ipfschain-
model@0.0.1.bna
```

2. Menjalankan jaringan bisnis.

```
composer network start --networkName ipfschain-model --networkVersion 0.0.1
--networkAdmin admin --networkAdminEnrollSecret adminpw --card
PeerAdmin@hlfv1 --file networkadmin.card
```

3. Mengimpor identitas administrator jaringan sebagai kartu jaringan bisnis.

```
composer card import --file networkadmin.card
```

4. Memeriksa apakah jaringan bisnis berhasil disebarkan

```
composer network ping --card admin@ipfschain-model
```

4.1.4 Membangun REST SERVER API (*middleware*)

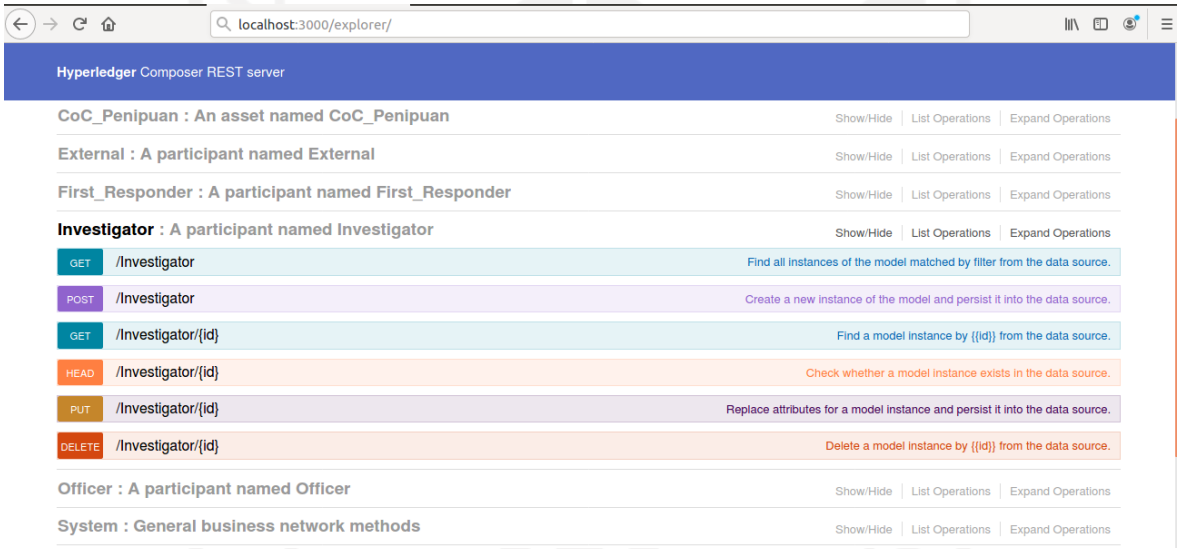
Hyperledger composer dapat menghasilkan REST API untuk mengembangkan aplikasi web. Fungsi REST SERVER API ini adalah sebagai penghubung antara sisi admin dengan sisi klien. Informasi terlihat pada gambar 4.6. Kemudian, membuat REST API navigasikan ke direktori ipfschain-model dan jalankan perintah seperti di bawah. Apabila berhasil dilakukan, maka dapat dibuka pada browser dengan menempelkan *http://localhost:3000/explorer.* Selanjutnya ditunjukkan pada gambar 4.7.

1. Buat REST API dengan menjalankan perintah:

```
composer-rest-server
```

2. Nama kartu: admin@ipfschain-model.
3. Apakah menggunakan ruang nama di API yang dihasilkan: *never use namspace.*
4. Apakah API yang dihasilkan diamankan: *no.*
5. Apakah akan mengaktifkan publikasi *event*: *yes.*
6. Apakah keamanan TLS diaktifkan: *no.*

Gambar 4.6 *Web playgorund*, *middleware*, dan *angular web app*



Gambar 4.7 REST API

4.1.5 Membangun Angular Web App (*fornt-end*)

Setelah melakukan langkah di atas, maka API yang dihasilkan terhubung ke *blockchain* dan jaringan bisnis yang digunakan. Selanjutnya, pada tahap ini membuat aplikasi kerangka angular 4 dengan memanfaatkan *tool hyperledger composer* yang berjalan dengan REST API yang telah dibuat. Apabila REST API tidak aktif, maka tidak bisa membuat atau menjalankan aplikasi angular. Berikut tahapannya:

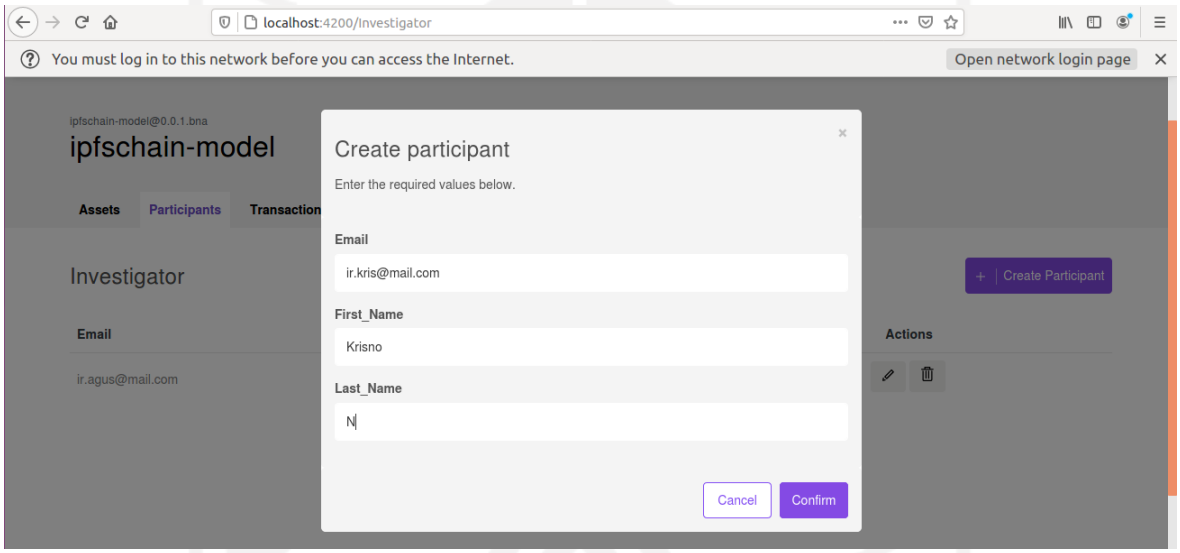
1. Buat aplilkasi angular di direktori ipfschain-model.

```
yo hyperledger-composer:angular
```

2. Apakah terhubung dengan jaringan bisnis: *yes.*

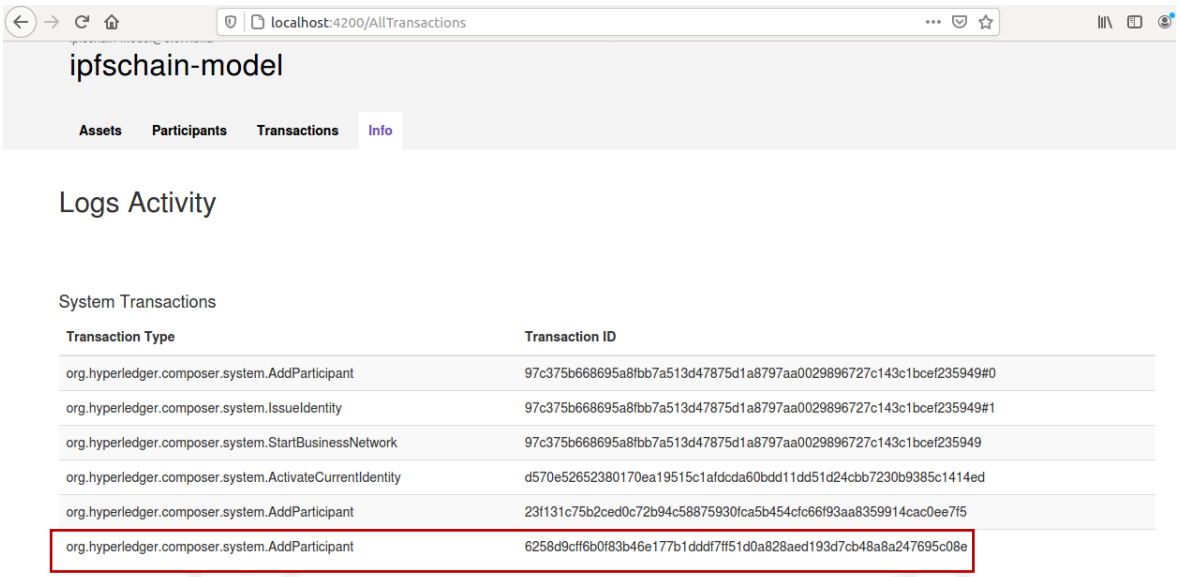
3. Informasi berupa *project name, description, author name, author email, licensi* disesuaikan dengan metadata yang ada di file *package.json.*
4. Pilih *Connect to an existing REST API.*
5. Alamat REST server: *http://localhost.*
6. *Port*: 3000.
7. Pilih: *Namespaces are not used.*

Generator angular akan membuat dan menginstal semua dependensi. Selain itu, ini ditujukan untuk menghasilkan *skeleton web application* berdasarkan defenisi model yang telah dibuat sebelumnya. Adapun model tampilan antar muka diberikan pada gambar dibawah ini.



Gambar 4.8 Tampilan halaman *create participant*

Gambar 4.8 sebagai representasi tampilan menu lainnya. Apabila tambah partisipan berhasil dilakukan, maka *log activity* dapat dilihat melalui menu info seperti yang ditunjukkan pada gambar 4.9.



Gambar 4.9 Tampilan halaman *logs activity*

4.2 Pengujian Sistem

Pengujian sistem IPFSChain dilakukan secara paralel yang terdiri dari pengujian pada konsep *off-chain* IPFS dan konsep *on-chain* HF. Pengujian dilakukan pada sisi admin melalui *web playgorund.* Secara konsep apakah alur sistem dan hak akses atau otoritas berjalan sesuai dengan perancangan yang dilakukan sebelumnya. Adapun pengujian sistem ini dilakukan pada aspek-aspek fungsionalitas, implementasi konsep DEC pada IPFSChain, dan kinerja. Selanjutnya, pengujian pada kinerja sistem dilakukan satu kali percobaan (*round*). Adapun pengujian untu beberapa kali percobaan dipaparkan pada penelitian (Hanafi et al., 2021) yang berfokus pada nilai *throughput* dan *latency*. Sedangkan hasil pengujian akan dilakukan secara sekuensial untuk uji banding dengan hasil uji pada penelitian yang sudah ada.

4.2.1 Pengujian Fungsionalitas IPFSChain

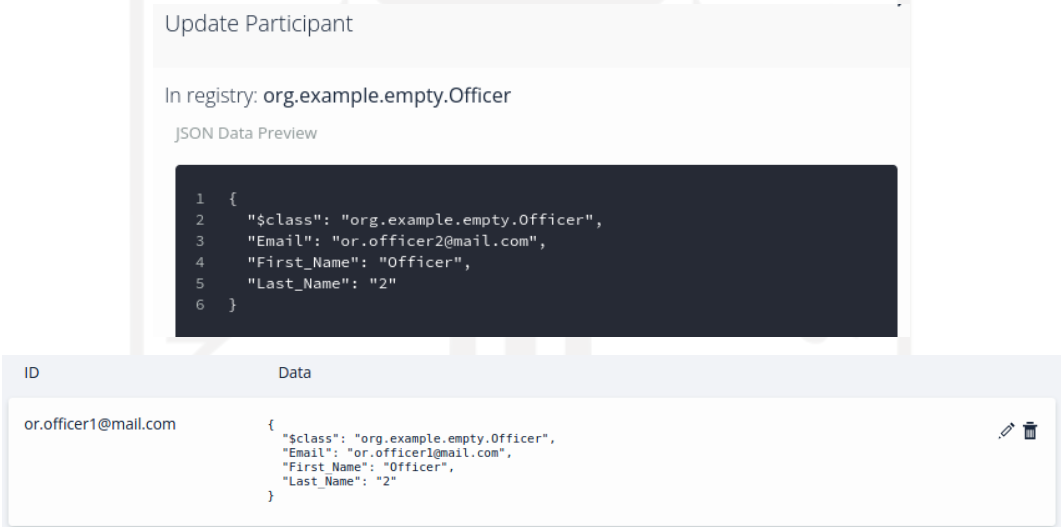
Pengujian fungsionalitas dilakukan untuk mengetahui apakah otoritas masing-masing pihak (*stakeholders*) berjalan sesuai dengan perancangan. Pengujian ini mengacu pada table 3.6 dan hasil dari pengujian untuk kesesuaian perbandingan pada tabel 4.2. Adapun tabel 4.2 mengacu pada tabel 3.1, 3.2, dan 3.3. Di bawah ini beberapa ototritas partisipan yang diuji.

A. Pengujian Fungsionalitas *Officer* (admin)

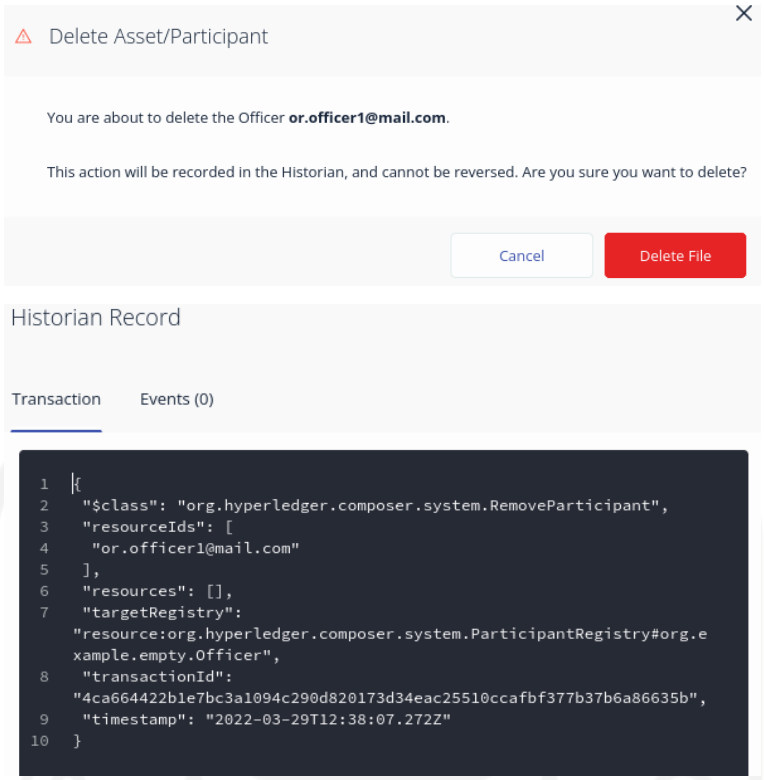
Berikut adalah hasil pengujian fungsionalitas terhadap otoritas *officer (admin).*

1. Melakukan *Create, Read, Update, Delete* pada *Officer*

Gambar 4.10 *Admin* menambah dan melihat anggota pada pihak *officer*



Gambar 4.11 *Admin* membarui anggota pada pihak *officer*

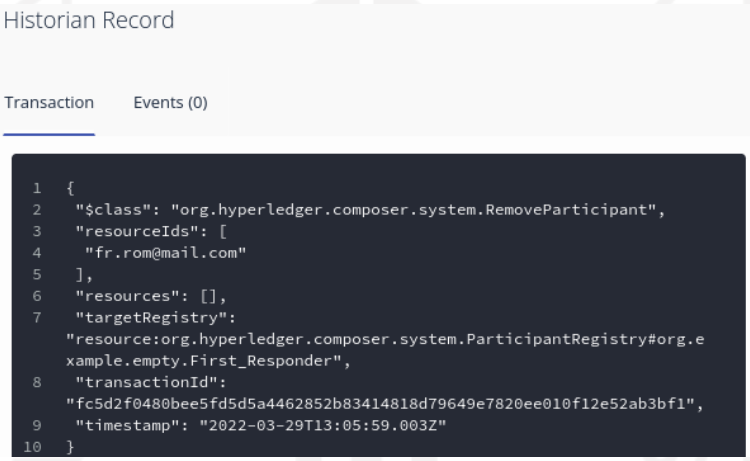


Gambar 4.12 *Admin* menghapus anggota pada pihak *officer*

2. Melakukan *Create, Read, Update, Delete* pada *First Responder*

Gambar 4.13 *Admin* menambah dan melihat anggota pada pihak *first responder*

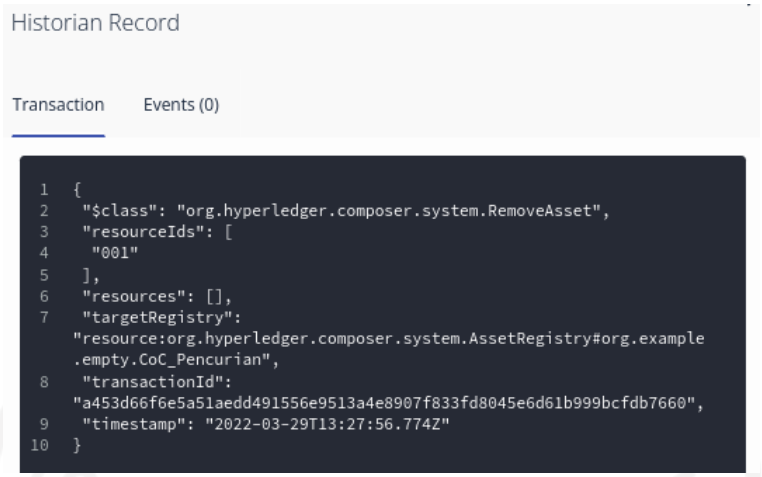
Gambar 4.14 *Admin* membarui anggota pada pihak *first responder*



Gambar 4.15 *Admin* menghapus anggota pada pihak *first responder*

Gambar di atas menunjukkan bahwa anggota dari pihak officer dapat melakukan create, read, update, dan delete pada anggota dari pihak first responder. Sehingga pengujian tersebut telah sesuai dengan perancangan.

3. Melakukan *Read*, *Delete* pada aset CoC_Pencurian

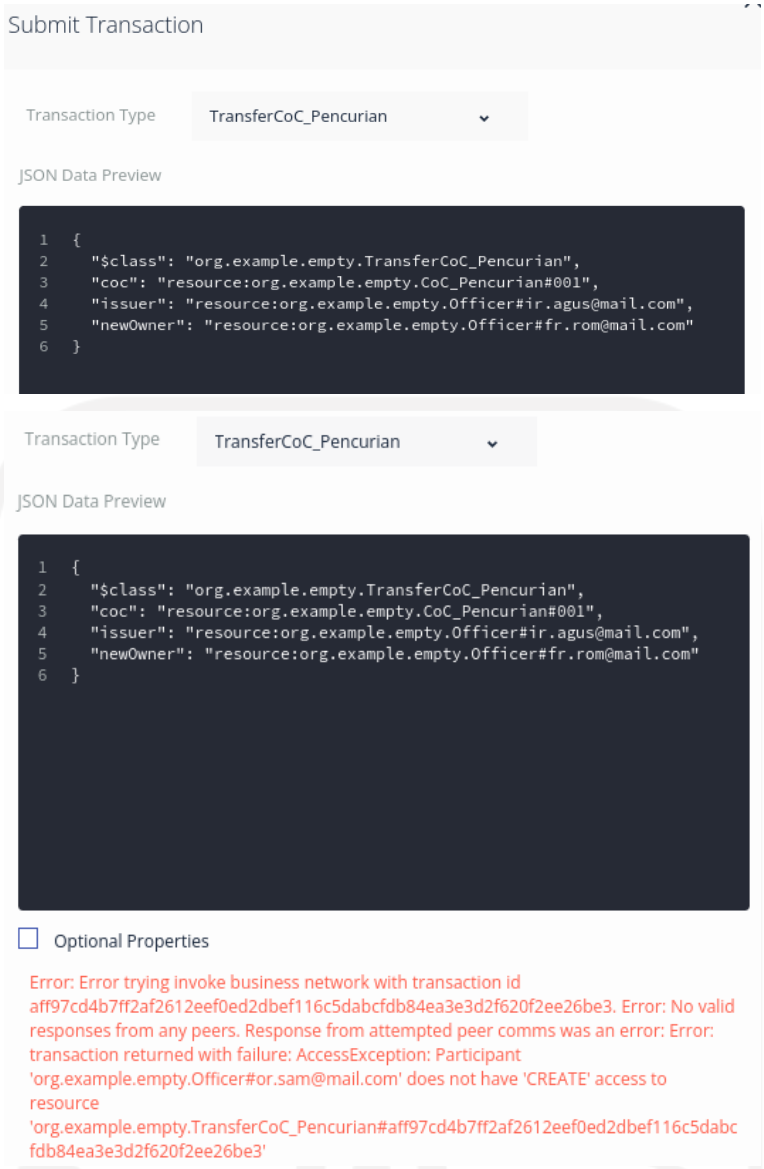


Gambar 4.16 *Admin* melihat dan menghapus aset CoC_Pencurian

Gambar 4.16 menunjukkan bahwa anggota dari pihak officer dapat melihat dan menghapus aset dari rak CoC_Pencurian. Sehingga pengujian tersebut telah sesuai dengan perancangan.

4. Melakukan *Create* pada *transfer* CoC_Pencurian

   Otoritas untuk submit transaksi berupa transfer salah satu aset dari rak CoC_Pencurian, tidak bisa dilakukan oleh pihak manapun selain pihak *investigator* yang telah diberikan izin.



Gambar 4.17 *Admin* submit transaksi berupa transfer aset CoC_Pencurian

Gambar 4.17 menunjukkan bahwa anggota dari pihak *officer* tidak dapat melakukan submit transaksi berupa transfer aset CoC. Sehingga pengujian telah sesuai dengan perancangan.

B. Pengujian Fungsionalitas *First Responder*

Berikut adalah hasil pengujian fungsionalitas terhadap otoritas *first responder*.

1. Melakukan *Read* pada *Officer*

Gambar 4.18 *First responder* melihat anggota dari pihak *offcier*

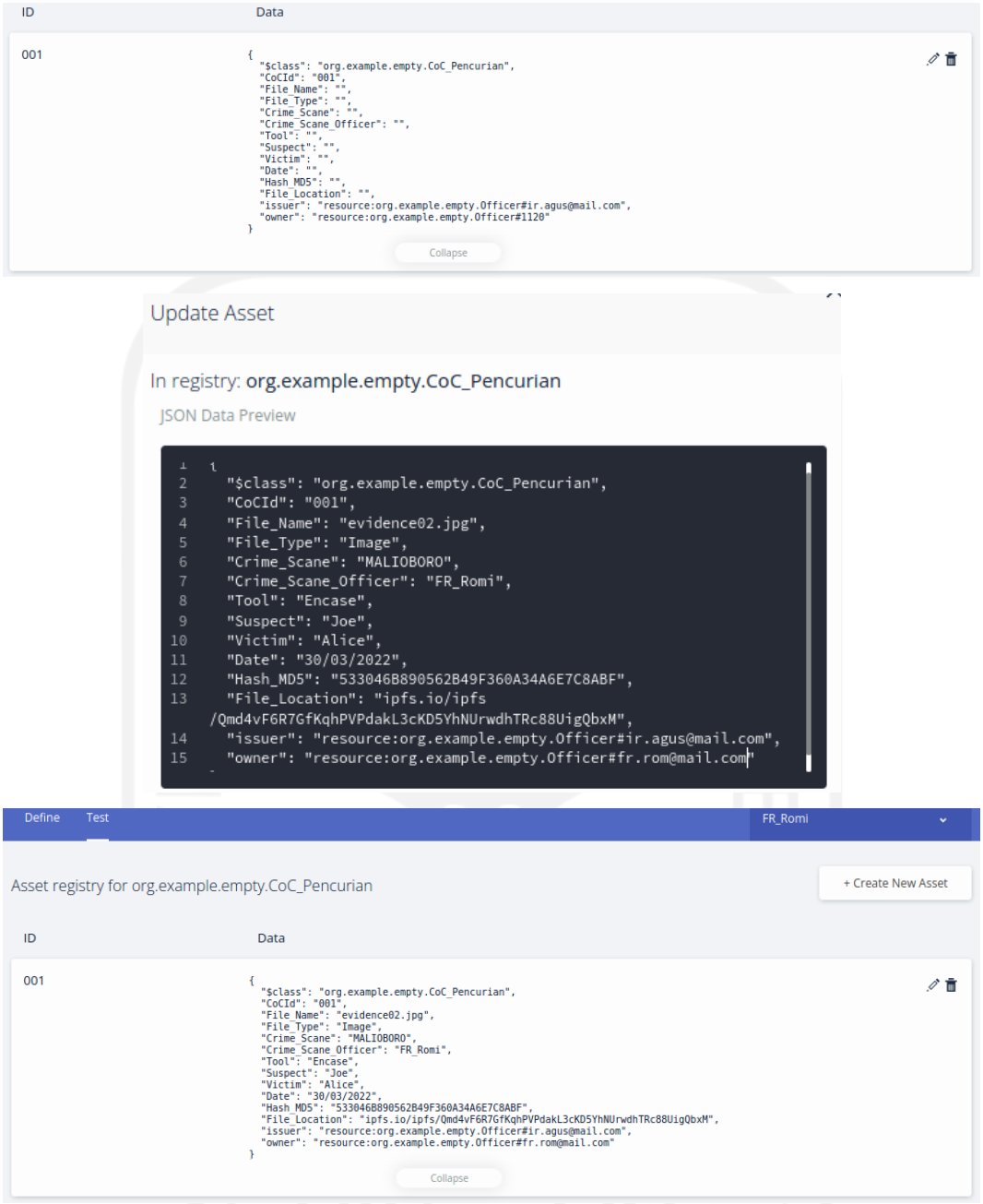
Gambar 4.18 menunjukkan bahwa salah satu anggota dari pihak *first responder* dapat melihat anggota dari pihak *officer*. Sehingga pengujian tersebut telah sesuai dengan perancangan.

2. Melakukan *Read* pada *Investigator*

Gambar 4.19 *First responder* melihat anggota dari pihak *investigator*

Gambar 4.19 menunjukkan bahwa salah satu anggota dari pihak *first responder* dapat melihat anggota dari pihak *investigator*. Sehingga pengujian tersebut telah sesuai dengan perancangan.

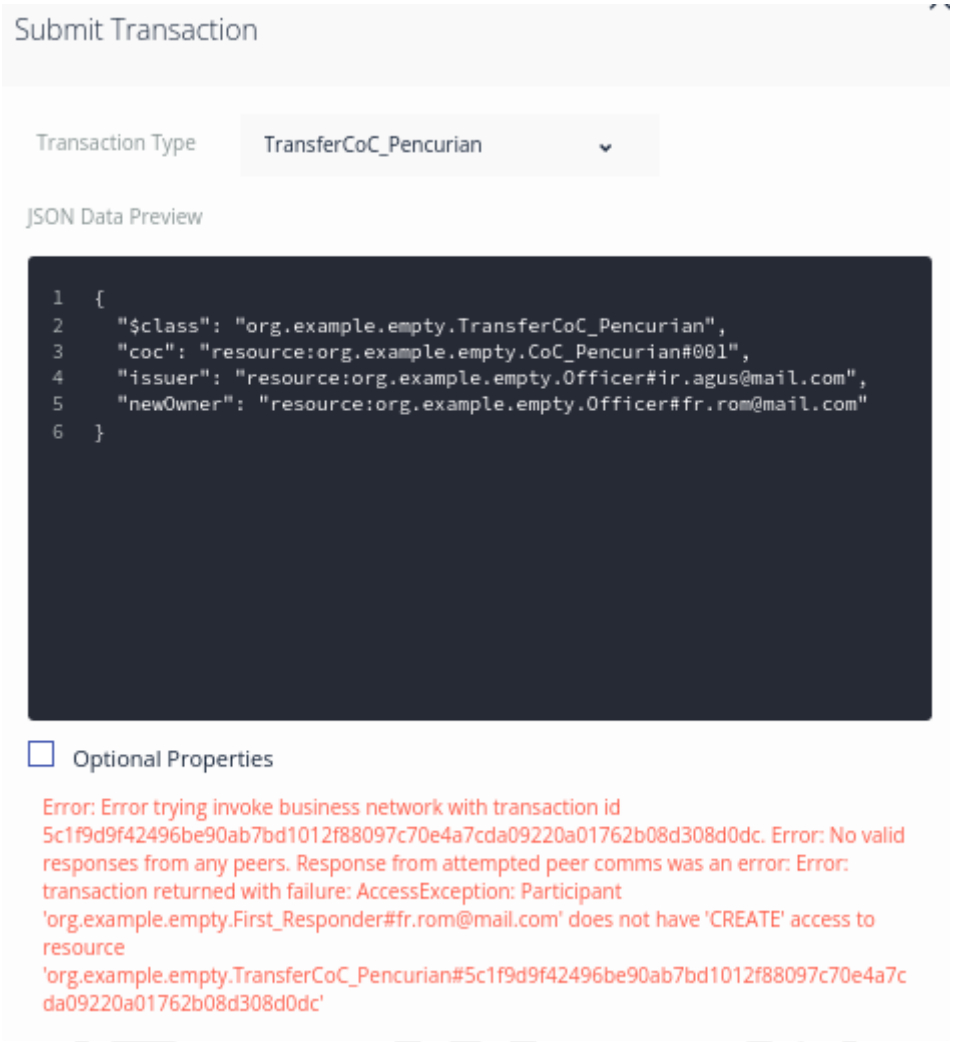
3. Melakukan *Read*, *Update* pada Aset CoC_Pencurian



Gambar 4.20 *First responder* melihat dan membarui aset CoC_Pencurian

Gambar 4.20 menunjukkan bahwa salah satu anggota dari pihak *first responder* dapat melihat dan memperbarui aset yang terdapat pada rak CoC_Pencurian. Sehingga pengujian tersebut telah sesuai dengan perancangan.

4. Melakukan *Create* pada *transfer* CoC_Pencurian



Gambar 4.21 *First responder* submit transaksi berupa transfer aset CoC_Pencurian

Gambar 4.21 menunjukkan bahwa anggota dari pihak *first responder* tidak dapat melakukan submit transaksi berupa transfer aset CoC. Sehingga pengujian telah sesuai dengan perancangan.

C. Pengujian Fungsionalitas *Investigator*

Berikut adalah hasil pengujian fungsionalitas terhadap otoritas *investigator*.

1. Melakukan *Read* pada *Officer*

Gambar 4.22 *Investigator* melihat anggota dari pihak *offcier*

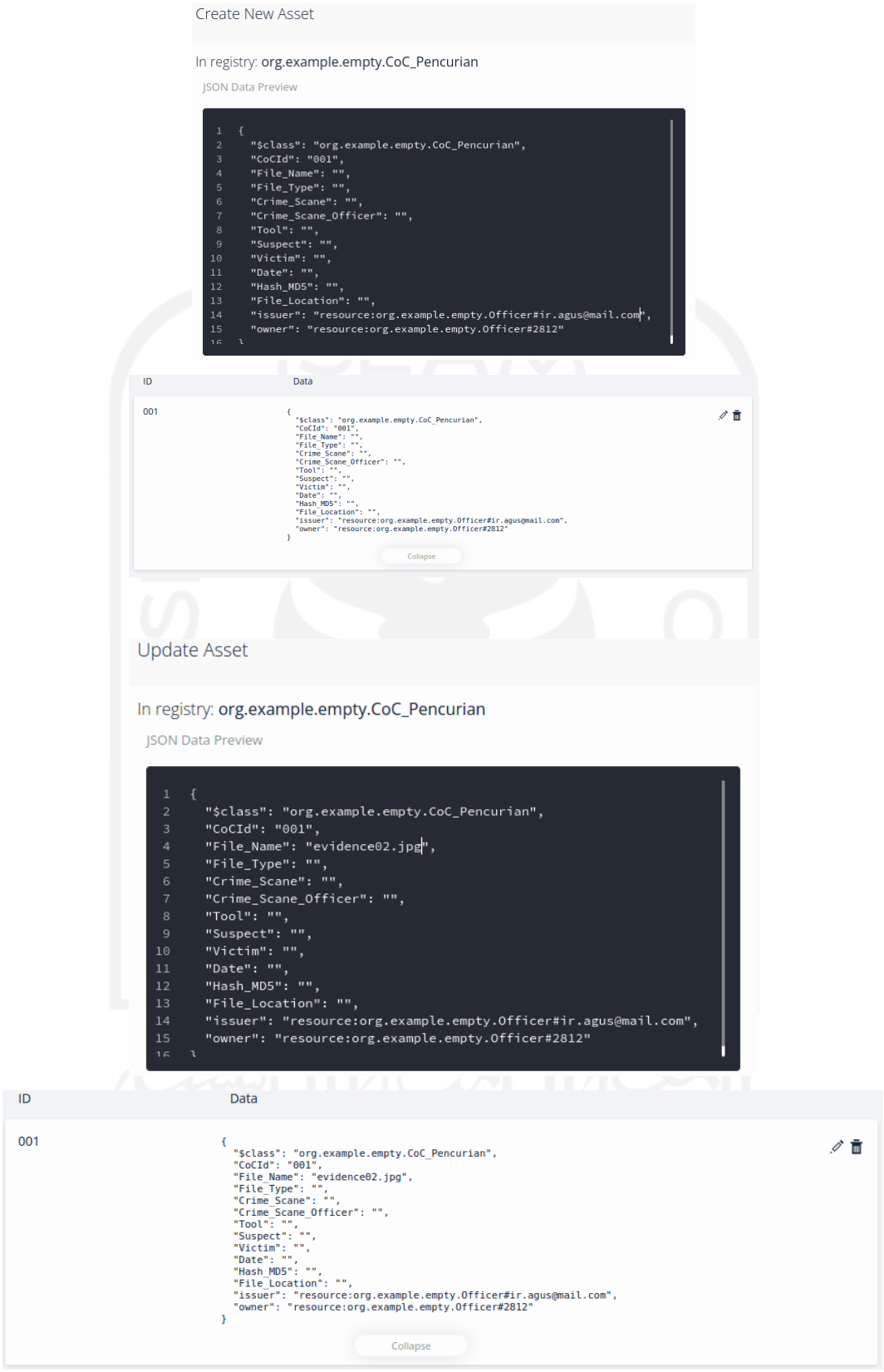
Gambar 4.22 menunjukkan bahwa salah satu anggota dari pihak *investigator* dapat melihat anggota dari pihak *officer*. Sehingga pengujian tersebut telah sesuai dengan perancangan.

2. Melakukan *Read* pada *First Responder*

Gambar 4.23 *Investigator* melihat anggota dari pihak *offcier*

Gambar 4.23 menunjukkan bahwa salah satu anggota dari pihak *investigator* dapat melihat anggota dari pihak *first responder*. Sehingga pengujian tersebut telah sesuai dengan perancangan.

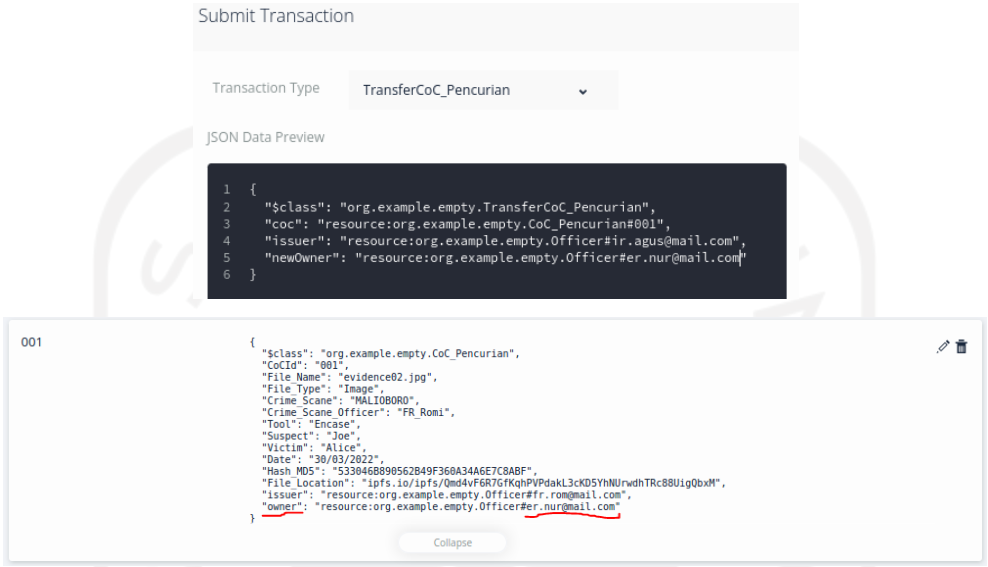
3. Melakukan *Create*, *Read*, *Update* pada Aset CoC_Pencurian



Gambar 4.24 *Investigator* membuat, melihat dan membarui aset CoC_Pencurian

Gambar 4.24 menunjukkan bahwa salah satu anggota dari pihak *investigator* dapat melakukan *create*, *read*, dan *update* aset. Sehingga pengujian tersebut telah sesuai dengan perancangan.

4. Melakukan *Create* pada *transfer* CoC_Pencurian



Gambar 4.25 *Investigator* submit transaksi berupa transfer aset CoC_Pencurian

Gambar 4.25 menunjukkan bahwa salah satu anggota dari pihak *investigator* dapat melakukan submit transaksi berupa transfer aset CoC kepada anggota dari pihak external. Sehingga status kepemilikan aset tersebut telah diperbarui. Adapun pengujian tersebut telah sesuai dengan perancangan.

D. Pengujian Fungsionalitas *External*

Berikut adalah hasil pengujian fungsionalitas terhadap otoritas *external*.

1. Melakukan *Read* pada *First Responder*

Gambar 4.26 *External* melihat anggota dari pihak *first responder*

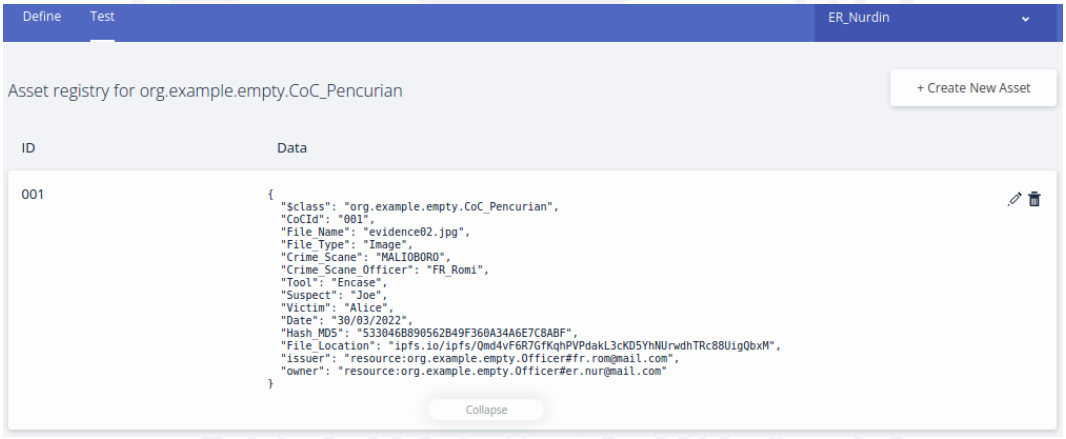
Gambar 4.26 menunjukkan bahwa anggota dari pihak *external* dapat melihat anggota dari pihak *first responder*. Sehingga pengujian telah sesuai dengan perancangan.

2. Melakukan *Read* pada *Investigator*

Gambar 4.27 *External* melihat anggota dari pihak *investigator*

Gambar 4.27 menunjukkan bahwa anggota dari pihak *external* dapat melihat anggota dari pihak *investigator*. Sehingga pengujian telah sesuai dengan perancangan.

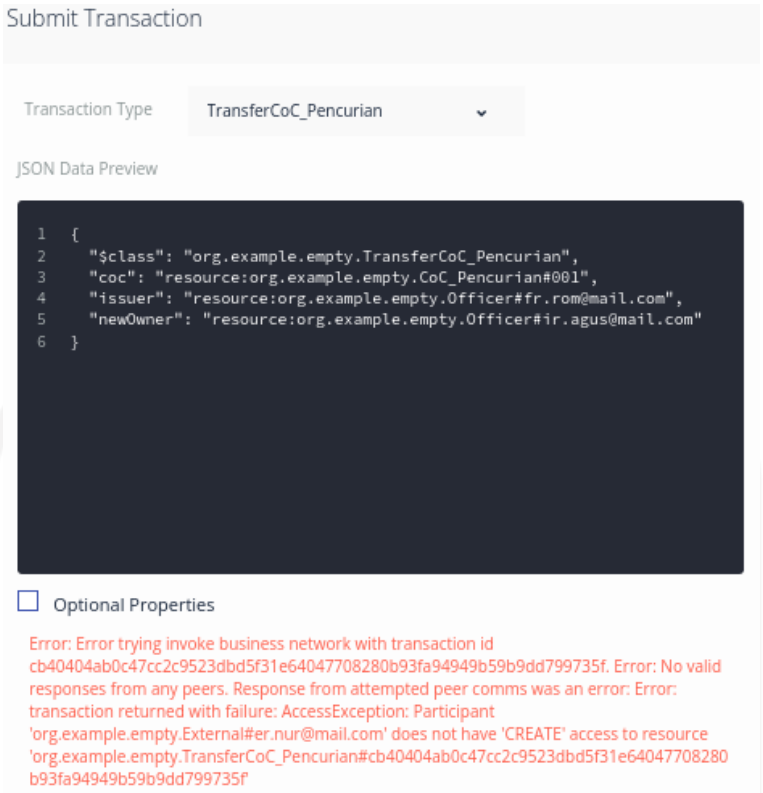
3. Melakukan *Read* pada Aset CoC_Pencurian



Gambar 4.28 *External* melihat aset CoC_Pencruian

Gambar 4.28 menunjukkan bahwa anggota dari pihak *external* dapat melihat aset yang disimpan dalam rak CoC_Pencurian. Sehingga pengujian telah sesuai dengan perancangan.

4. Melakukan *Create* pada Transfer CoC_Pencurian



Gambar 4.29 *External* submit transaksi berupa transfer aset CoC_Pencruian

Gambar 4.29 menunjukkan bahwa anggota dari pihak *external* tidak dapat melakukan submit transaksi berupa transfer aset CoC. Sehingga pengujian telah sesuai dengan perancangan.

Tabel 4.2 Daftar skenario pengujian fungsionalitas

No	Partisipan	Aktivitas	Izin	Keterangan
1	Officer (Admin)	Melakukan *Create, Read, Update, Delete* pada *Officer*	Bisa	Sesuai
		Melakukan *Create, Read, Update, Delete* pada *First Responder*	Bisa	Sesuai
		Melakukan *Create, Read, Update, Delete* pada *Investigator*	Bisa	Sesuai
		Melakukan *Create, Read, Update, Delete* pada *External*	Bisa	Sesuai
		Melakukan *Read, Delete* pada aset CoC_Pencurian	Bisa	Sesuai
		Melakukan *Read, Delete* pada aset CoC_Penghinaan	Bisa	Sesuai
		Melakukan *Read, Delete* pada aset CoC_Penipuan	Bisa	Sesuai
		Melakukan *Create* pada *Transfer* CoC_Pencurian	Tidak Bisa	Sesuai
2	First Responder	Melakukan *Read* pada *Officer*	Bisa	Sesuai
		Melakukan *Read* pada *First Responder*	Bisa	Sesuai
		Melakukan *Read* pada *Investigator*	Bisa	Sesuai
		Melakukan *Read* pada *External*	Bisa	Sesuai
		Melakukan *Read, Update* pada Aset CoC_Pencurian	Bisa	Sesuai
		Melakukan *Read, Update* pada Aset CoC_Penghinaan	Bisa	Sesuai
		Melakukan *Read, Update* pada Aset CoC_Penipuan	Bisa	Sesuai
		Melakukan *Create* pada *Transfer* CoC_Pencurian	Tidak Bisa	Sesuai
3	Investigator	Melakukan *Read* pada *Officer*	Bisa	Sesuai
		Melakukan *Read* pada *First Responder*	Bisa	Sesuai
		Melakukan *Read* pada *Investigator*	Bisa	Sesuai
		Melakukan *Read* pada *External*	Bisa	Sesuai
		Melakukan *Create, Read, Update* pada Aset CoC_Pencurian	Bisa	Sesuai
		Melakukan *Create, Read, Update* pada Aset CoC_Penghinaan	Bisa	Sesuai

		Melakukan *Create, Read, Update* pada Aset CoC_Penipuan	Bisa	Sesuai
		Melakukan *Create* pada Transfer CoC_Pencurian	Bisa	Sesuai
4	External	Melakukan *Read* pada *Officer*	Bisa	Sesuai
		Melakukan *Read* pada *First Responder*	Bisa	Sesuai
		Melakukan *Read* pada *Investigator*	Bisa	Sesuai
		Melakukan *Read* pada *External*	Bisa	Sesuai
		Melakukan *Read* pada Aset CoC_Pencurian	Bisa	Sesuai
		Melakukan *Read* pada Aset CoC_Penghinaan	Bisa	Sesuai
		Melakukan *Read* pada Aset CoC_Penipuan	Bisa	Sesuai
		Melakukan *Create* pada *Transfer* CoC_Pencurian	Tidak Bisa	Sesuai

Tabel 4.2 menunjukkan bahwa pengujian fungsionalitas dapat dilakukan dan mendapatkan hasil yang sesuai dengan perancangan.

4.2.2 Pengujian Implementasi IPFSChain

Tahap ini dilakukan pengujian terhadap *chaincode* untuk memastikan kemampuan sistem IPFSChain apakah sesuai dengan alur proses sistem yang sudah ada sebelumnya.

Tabel 4.3 Pengujian Implementasi IPFSChain

No	Proses	B-DEC	Multi Smart Contract	IPFSChain
1	Login sistem Blockchain	✓	✓	-
2	Penyimpanan data user dan CoC ke database	✓	✓	✓
3	Penyimpanan metadata dan/atau bukti digital ke dalam sistem	✓	✓	✓
4	Penyimpanan bukti digital ke dalam sistem penyimpanan distributable (IPFS)	-	-	✓
5	Penyimpanan Hash bukti digital	✓	✓	✓
6	Penyimpanan Log aktivitas terhadap bukti digital dan *chain ofcustody*	✓	✓	✓
7	Ekstraksi informasi dengan GetID3 untuk meningkatkan integritas bukti digital	-	✓	-
8	Input informasi berdasarkan kaidah CoC	✓	-	✓
9	Pengguna tersertifikasi oleh sistem *blockchain* (*digital signature*)	-	-	✓

Dari tabel 4.3 terlihat bahwa, konsep digital CoC berbasis blockchain yang digunanakan secara prinsip telah sesuai dengan penelitian sebelumnya. Namun dengan menerapkan IPFS sebagai penyimpanan yang distributable, memberikan tingkat kinerja sistem yang efektif dan efisien.

4.2.3 Pengujian Kinerja IPFSChain

Pengujian kinerja IPFSChain ini terdiri dari uji sistem pada IPFS (*off-chain*) dan uji sistem pada HF (*on-chain*). Tujuan uji kinerja adalah untuk mengukur kemampuan sistem dalam melakukan proses terhadap suatu data dan aktivitas *user*. Selain itu, untuk mengetahui seberapa efektif dan efisien konsep IPFSChain model ini diterapkan.

Tabel 4.4 Pengujian kinerja IPFS

No	Digital Evidence	Response Time (s)		Size (KB)
		Add	Get	
1	evidence01.txt	0.04	0.04	5
2	evidence02.jpg	0.04	0.04	50
3	evidence03.pdf	0.15	0.34	500
4	evidence04.mp3	1.42	2.45	5000
5	evidence05.mp4	6.96	18.73	50000

Dari tabel 4.3 di atas dapat dipahami bahwa, waktu akses/unduh ke dalam sistem IPFS lebih lama dari pada waktu unggah. Ini karena sistem IPFS memecah menjadi beberapa data (*Merkle-DAG Concept*) sebelum menyimpan data ke dalam sistemnya. Selanjutnya, sistem menyatukan kembali potongan data tadi menjadi satu-kesatuan yang utuh. Karena itu unduh data secara utuh membutuhkan waktu eksekusi yang lebih lama.

Jika uji skema pada tabel 4.3 ada 5 untuk masing-masing aktivitas yaitu *add* dan *get,* maka total keseluruhan uji skema menjadi 10 dan *size* masing-masing skema di x 2. Sehingga total ukuran data pada kolom *size* adalah 111110 KB. Apabila dibandingkan dengan penelitian sebelumnya, yang pertama dalam hal ini diperlukan nilai rata-rata dari ukuran file. Adapun rumus yang digunakan seperti dibawah ini.

$$\overline{Size} = \sum \frac{Size}{Skema} \tag{4.1}$$

Dengan rumus ini diperoleh nilai rata-rata ukuran file adalah 111110 KB dibagi 10 menjadi 11111 KB. Selanjutnya menghitung nilai rata-rata waktu dengan rumus berikut.

$$\overline{Time} = \sum \frac{Time}{Skema} \tag{4.2}$$

Jika kedua aktivitas *add* dan *get* tersebut digabungkan, maka total waktu yang diperoleh menjadi 30.21 detik. Apabila nilai tersebut dimasukkan ke dalam rumus, maka 30.21 detik dibagi 10 menjadi 3.021 detik. Selanjutnya menentukan nilai rasio dengan rumus berikut.

$$Rasio = \frac{\overline{Size}}{\overline{Time}} \quad (4.3)$$

Untuk menentukan nilai rasio adalah dengan menghitung nilai rata-rata ukuran file dibagi dengan nilai rata-rata waktu, maka nilai 11111 KB dibagi 3.021 detik menjadi 3678 KB/s. Namun rasio tersebut adalah patokan nilai pada sisi *off-chain.* Adapun penghitungan nilai rasio pada sisi *on-chain* tetap menggunakan rumus yang sama seperti di atas.

Pengujian pada sisi *on-chain* HF menggunakan *tool* JMeter yang diinstall pada sistem operasi ubuntu yang berjalan menggunakan *VirtualBox*. Percobaan dilakukan dengan *http request* pada *port* 3000 *rest server api.* Aktivitas tersebut adalah *post/create* dan *delete* pada masing-masing komponen. Adapun hasil pengujian terhadap partisipan diuraikan pada tabel 4.4, pengujian terhadap aset diuraikan pada tabel 4.5, dan pengujian terhadap transfer aset diuraikan pada tabel 4.6.

Tabel 4.5 Pengujian pada komponen partisipan

Label	Avg. Latency (s)	Throughput (s)	Received (KB/s)	Sent (KB/s)	Size (KB)
Post External	2.769	0.361	0.18	0.122	0.499
Post First Responder	2.906	0.344	0.163	0.108	0.474
Post Investigator1	2.720	0.368	0.174	0.114	0.472
Post Investigator2	2.815	0.355	0.169	0.111	0.474
Post Officer	2.693	0.371	0.174	0.111	0.467
Delete External	3.002	0.333	0.105	0.065	0.315

Tabel 4.6 Pengujian pada komponen aset

Label	Avg. Latency (s)	Throughput (s)	Received (KB/s)	Sent (KB/s)	Size (KB)
POST-CoC_Pencurian1	2.951	0.339	0.284	0.246	0.836
POST-CoC_Pencurian2	2.991	0.334	0.282	0.245	0.843
POST-CoC_Penghinaan1	2.651	0.377	0.319	0.277	0.844
POST-CoC_Penghinaan2	2.949	0.339	0.287	0.249	0.846
POST-CoC_Penipuan	3.220	0.311	0.26	0.225	0.836
DELETE-CoC_Penipuan	2.998	0.334	0.105	0.063	0.315

Tabel 4.7 Pengujian pada komponen transfer

Label	Avg. Latency (s)	Throughput (s)	Received (KB/s)	Sent (KB/s)	Size (KB)
POST - 1 Transfer Aset Pencurian	2.683	0.373	0.253	0.165	0.677

Dari ketiga tabel diatas, bahwa rata-rata *latency* atau waktu yang dibutuhkan untuk memproses suatu permintaan berhasil dilakukan, tanpa terjadi *corupt* atau *error* adalah 2.872 detik per label/aktivitas. Adapun ringkasan dari ketiga tabel 4.5, 4.6, dan 4.7 akan diuraikan

pada tabel 4.8. Tujuannya adalah untuk memudahkan dalam tahapan perbandingan kinerja sistem IPFSChain dengan penelitian sebelumnya.

Tabel 4.8 Pengujian kinerja HF

No	Components	Activities	Size (KB)	Avg. Latency (s)
1	Partisipan	Tambah 1 anggota *officer*	0.467	2.693
		Tambah 1 anggota *first responder*	0.474	2.906
		Tambah 2 anggota *investigator*	0.946	5.535
		Tambah 1 anggota *external*	0.499	2.769
		Hapus 1 anggota dari salah satu pihak	0.315	3.002
2	Aset CoC	Tambah 2 aset ke dalam rak CoC_Pencurian	1.679	5.942
		Tambah 2 aset ke dalam rak CoC_Penghinaan	1.690	5.600
		Tambah 1 aset ke dalam rak CoC_Penipuan	0.836	3.220
		Hapus 1 aset dari salah satu rak	0.315	2.998
3	Transfer CoC	Submit transaksi 1 aset	0.677	2.683

Selanjutnya dari tabel 4.8 dilakukan pengujian dengan menggunakan ketiga rumus di atas. Adapun rata-rata dari ukuran file secara keseluruhan adalah 0.789 KB. Untuk nilai rata-rata latency adalah 3.734 detik. Maka untuk mendapatkan nilai rasio 0.789 KB dibagi 3.734 detik menjadi 0.211 KB/s. Selanjutnya dibahas dan diuraikan pada tabel 4.10.

4.3 Analisis Sistem

4.3.1 Analisis Implementasi IPFSChain

Analisis implementasi sistem IPFSChain ini akan dibahas pada dua sisi, yaitu sisi admin dan sisi klien. Sisi admin akan dianalisis melalui *web plyaground* atau *composer-playground* yang menggunakan *port* 8080. Sedangkan analisis pada sisi klien melalui *angular web app* dengan *port* 4200. Analisis pada sisi klien tidak dilakukan secara kompleks, karena pada penelitian ini juga melakukan percobaan implementasi konsep *Digital Evidence Cabinet* (DEC). Bahwa implementasi konsep rak pada DEC dapat diterapkan pada sisi klien dan juga dapat melakukan semua fungsi *create, read, update,* dan *delete*. Selain itu juga dapat melihat semua aktivitas yang dilakukan peserta terhadap sistem IPFSChain melalui menu info yang ada di *angular web app* tersebut. Seperti yang terlihat pada gambar 4.9. Akan tetapi itu semua dapat dilakukan, hanya jika *user* yang diberikan link dan waktu aksesnya kapan.

Namun kekurangannya pada sisi klien ini, belum dapat diterapkan akses kontrol seperti halnya pada *web-playground* yang digunakan admin. Selain itu, *angular web app* belum diterapkan sistem *login* untuk keamanan. Adapun analisis yang dilakukan secara garis besar pada sisi admin ditunjukkan pada tabel 4.9.

Tabel 4.9 Hasil pengujian IPFSChain

No	Klausul	B-DEC	Multi Smart Contract	IPFSChain	Keterangan
1	*Authentication* pada halaman *login*	✓	✓	-	*Input username* dan *password* saat *login*
2	Mampu mengakomodir lebih dari satu bukti digital	✓	✓	✓	Dalam satu kasus memiliki banyak bukti digital
3	Informasi bukti digital bersifat dinamis	✓	-	✓	Informasi dientri susai dengan kaidah CoC atau bersifat dinamis
4	*Smart contract* dinamis sesuai kebutuhan	-	✓	✓	Menyimpan metadata ke blok berdasarkan kasus
5	Bukti digital disimpan dalam sistem IPFS yang memiliki varian jenis dan ukuran	-	-	✓	File asli hanya dapat diakses menggunakan *multi-hash* dari IPFS
6	Memiliki *log* aktivitas terhadap aset	✓	✓	✓	Aksi *input*, *update*, *read*, dan *delete* disimpan dalam blok
7	Partisipan tersertifikasi sebagai *signature* secara digital	-	-	✓	*Card* ID berupa *key private* dan *certification authority* dari sistem HF

Dari tabel 4.9 terdapat perbedaan kurang lebih tiga klausul, yang pertama terkait sistem *login*. Sistem *login* pada IPFSChain tidak diterapkan, karena konfigurasi yang sangat kompleks untuk membuat sistem tokenisasi (token) sebagai ganti *username* dan *password* untuk keamanan seperti yang digunakan aplikasi pada umumnya. Kedua, pada poin 3 sistem IPFSChain ini khususnya pada sisi *on-chain*, menjadikan aktivitas pencatatan dokumen CoC sebagai catatan bersama yang hanya bisa dilakukan oleh pihak *investigator* dan *first responder*. Adapun informasi yang dituliskan ke dalam aset dokumen CoC berdasarkan kaidah CoC yang diteliti oleh (Prayudi et al., 2014).

Namun kaidah tersebut masih dapat dilakukan pemutakhiran pada informasi yang terkandung dalam dokumen CoC tersebut dengan melakukan *upgrade versi* pada *chaincode*/*smart contract.* Sedangkan informasi yang disimpan dengan GetID3 dapat dikatakan tidak dinamis dan tidak efisien. Karena ketetapan aturan proses dalam sistem GetID3 dan semua informasi yang diekstrak disimpan ke sistem. Ketiga, pada poin ke 5 sistem IPFS dapat memberikan efektifitas dan efisien pada kinerja sistem. Adapun perbandingan datanya diuraikan pada tabel 4.11. Terkahir, pada poin 7 untuk masing-masing

pihak memiliki *cardID* sebagai anggota yang sah/valid dalam jaringan dan menjalankan fungsi sesuai dengan otoritas yang diberikan dan yang telah disepakti.

4.3.2 Analilsis Pengujian Fungsionalitas

Pada tahap ini akan membahas terkait partisipan sebagai komponen pada sistem yang perlu dideskripsikan. Hal ini bertujuan untuk memastikan semua kebutuhan dapat diakomodasi secara tepat dan jelas. Adapun analisis dilakukan pada pihak-pihak yang dibentuk dan otoritasnya dalam sistem IPFSChain sebagai berikut:

1. *Officer* (OR): sesuai dengan tugasnya bahwa pengguna ini adalah admin kedua yang diberikan otoritas tertentu oleh admin pertama (super admin) pada sistem. Adapun tugas dari anggota pada pihak ini adalah memiliki hak penuh atas partisipan atau dapat melakukan semua fungsi create, read, update, dan delete. Namun hanya dapat melihat dan menghapus aset yang terdapat dalam rak. Selain itu, tidak dapat melakukan submit transasksi berupa transfer terhadap aset.
2. *First Responder* (FR): anggota pada pihak ini yang berhubungan langsung dengan bukti. Sehinga anggota ini harus dapat diakomodir untuk melakukan entri informasi metadata dari suatu bukti ke dalam sistem HF dan mengunggah file bukti tersebut ke IPFS.
3. *Investigator* (IR): anggota ini yang bersentuhan langsung dengan bukti yang sebelumnya menerbitkan aset dengan mengisi nomor identitas aset dan identitas diri pada *form issuer*. Kemudian, melakukan submit transaksi berupa transfer aset kepada pihak lain yaitu pihak *external.*
4. *External* (ER): latar belakang anggota dari pihak ini dapat berasal dari berbagai instansi dan sejenisnya yang memiliki hak untuk akses terhadap CoC. Namun pada penelitian ini, anggota yang didaftarkan ke dalam pihak *external* pada sistem HF didefinisikan sebagai salah satu perwakilan jaksa.

Dari beberapa uraian di atas, dapat dirangkum bahwa beberapa anggota dari masing-masing pihak beserta dengan fasilitas yang diakomodir adalah sebagaimana yang terlihat pada tabel 4.10.

Tabel 4.10 Otoritas partisipan sistem IPFSChain

Participants	Activities	Actions
OR	Tambah anggota	*Create*
	Ubah informasi anggota	*Update*
	Hapus aset	*Delete*
FR	Unggah file bukti digital ke IPFS	*Add*
	Ubah informasi aset / pendokumentasian	*Update*
IR	Tulis aset	*Create*
	Ubah informasi aset	*Update*
	Submit transaksi berupa transfer aset	*Create*
ER	Unduh file bukti digital dari IPFS	*Get*
	Akses informasi partisipan, aset, dan *logs activity*	*Read*

4.3.3 Analisis Pengujian Kinerja

Pengujian kinerja pada IPFSChain dilakukan secara paralel karena terdiri dari dua konsep yang tidak diintegrasikan. Selain itu, data yang digunakan pada konsep IPFS dan HF memiliki ukuran dan jenis yang berbeda. Seperti yang terlihat pada tabel 3.4. Adapun penarikan kesimpulan pada hasil uji kinerja dilakukan secara sekuensial yang didefenisikan sebagai konsep model yang utuh yaitu, IPFSChain untuk pembanding dengan penelitian yang sudah ada.

Hasil pengujian kinerja sistem IPFSChain didapatkan suatu nilai rasio atau perbandingan dengan penelitian sebelumnya. Bahwa masing-masing konsep memiliki kelebihan dan kekurangan. Namun pada tahap ini dilakukan analisis kinerja sistem dan memaparkan informasi yang ditujukan untuk mengukur seberapa efektif dan efisien masing-masing implementasi sistem. Adapun penghitungan dan pengukuran pada peneliti (Putra & Prayudi, 2021) terkait *multi smart contract* terdapat kekeliruan pada saat menentukan hasil rasio dari *multi smart contract.* Dalam penelitiannya menggunakan rumus rasio yang terbalik, yaitu jumlah rata-rata waktu dibagi jumlah rata-rata ukuran data. Sehingga didapatkan hasil rasio menjadi 0.000991939. Berikut informasi yang dapat diuraikan pada tabel 4.11.

Tabel 4.11 Perbandingan kinerja IPFSChain, *Multi Smart Contract*, dan B-DEC

Klausul	IPFSChain		Multi Smart Contract	B-DEC
	IPFS	HF		
$\overline{Size}$	11111 KB	0.789 KB	2076.74 KB	0.97 KB
$\overline{Time}$	3.021 s	3.734 s	2.06 s	5.261 s
Rasio	3768 KB/s	0.211 KB/s	1008 KB/s	0.184 KB/s

Dari tabel di atas, bahwa nilai rasio pada sistem IPFSChain terbagi menjadi dua yaitu 3768 KB/s pada IPFS dan 0.211 KB/s pada HF. Ukuran data pada IPFS dan HF terdapat perbedaan yang cukup signifikan, karena sistem IPFS dirancang untuk menyimpan data yang

bervarian dan berukuran besar. Sedangkan pada sistem HF hanya digunakan untuk menyimpan data yang berukuran kecil berupa metadata karena bersifat permanen. Sebagaimana percobaan yang dilakukan pada penelitian (Hanafi et al., 2021) dengan komponen sistem HF yang terdiri dari 1 organisasi 4 rekan menunjukkan bahwa, peningkatan jumlah rekan dapat mengurangi *throughput* dan bertambahnya *latency.*

Selanjutnya, apabila kedua nilai tersebut dijumlahkan menjadi 3768.211 KB/s kemudian dibagi 2, maka hasilnya adalah 1884.1 KB/s. Apabila dibandingkan dengan dua sistem lainnya didapatkan rasio 2:1:0.0002. Maka, dengan data tersebut dapat disimpulkan bahwa sistem IPFSChain lebih efektif dan efisien. Dalam hal ini, efektif yang dimaksud adalah tolok ukur berdasarkan kecepatan proses sistem terhadap data dalam satuan KB/s. Adapun efisien adalah tolok ukur atas dasar skalabiliti pada volume penyimpanan dan informasi yang disesuaikan dengan kaidah CoC.

BAB 5
Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan paparan rancangan konsep DEC pada IPFSChain model berbasis *blockchain* berizin yaitu *Hyperledger* terhadap konten bukti digital dan CoC, maka dapat disimpulkan bahwa:

1. Perancangan konsep DEC pada IPFSChain model dalam manajemen bukti digital dan CoC dengan pendekatan *modular hyperledger composer* dan penyimpanan terdistribusi IPFS. Rancangan IPFSChain model atas dasar konsep *off-chain* dan *on-chain.* Perancangan meliputi proses analisis kebutuhan sistem, diagram alir sistem, desain arsitektur, desain *Business Network Archive* (BNA), dan desain antarmuka. *Output* dari perancangan IPFSChain model berupa alur penyimpanan konten bukti digital dan pendokumentasian CoC serta deskripsi komponen-komponen yang dibutuhkan untuk membangun IPFSChain dalam memberikan kemudahan dan keamanan akses dan transfer data dalam jaringan.
2. Implementasi IPFSChain model meliputi proses membangun konsep *off-chain* dan *on-chain.* Konsep *off-chain* dibangun sebagai sisi penyimpanan file bukti digital yang distributable. Sedangkan konsep *on-chain* dibangun sebagai sisi pendokumentasian dan penyimpanan CoC dalam bentuk metadata. Proses implementasi dimulai dari membangun sistem penyimpanan distributable IPFS, membangun *platform blockchain* HF, membangun *chaincode,* membangun *rest server api* sebagai *middleware,* membangun antarmuka, dan pengujian sistem. *Output* dari implementasi sistem berupa aplikasi berbasis web.
3. Pengujian implementasi sistem meliputi tahapan pengujian fungsionalitas, pengujian implementasi, dan pengujian kinerja sistem. Hasil pengujian sistem menunjukkan bahwa implementasi sistem secara prinsip sesuai dengan sistem-sistem yang sudah ada. Hasil pengujian kinerja pada sistem IPFSChain membuktikan bahwa secara keseluruhan aspek memberikan tingkat keefektifan dan efisien yang lebih baik. Selain itu juga meningkatkan kepercayaan antar pihak yang tergabung dalam jaringan.

5.2 Saran

Dari hasil yang diperoleh melalui penelitian ini menunjukkan bahwa, adanya kekurangan dan keterbatasan. Sehingga dibutuhkan beberapa hal yang perlu untuk dikembangkan dan diteliti lebih jauh yaitu:

1. Sistem IPFSChain ini belum dilengkapi dengan sistem autentikasi, sehingga diperlukan adanya pengembangan sistem dengan menerapkan strategi *passport-github* pada *rest server* untuk mengautentikasi pengguna/klien yang berupa token.
2. Sistem IPFSChain ini hanya diimplementasikan dan diuji pada satu mesin/perangkat. Karena itu perlu untuk dilakukan pengujian pada beberapa perangkat.
3. Mengkaji lebih jauh tentang sistem penyimpanan ditributable yang lebih efektif dan efisien, sehingga dapat diterapkan dengan sistem *Hyperledger* atau *blockchain* lainnya.

Daftar Pustaka

A., Douglas & Lancester, D., M. (2018). "Cryptocurrency and the Blockchain: A Discussion of Forensic Needs". *IJCSDF, 7 (4), p. 420-435.*

Ami-Narh, J.T., & Williams, P.A. (2008). "Digital forensics and the legal system: a dilemma of our times". In: Australian Digital Forensics Conference, vol. 41.

Andola, N. et al. (2019). "Vulnerabilities on Hyperledger Fabric". *Pevasive and Mobile Computing, 59.*

Androulaki, E. et al. (2018). "Hyerledger Fabric: A Distributed Operating System for Permissioned Blockchains". *Proceedings of the 13th, EuroSys Conference,* https://arxiv.org/pdf/1801.10228.pdf

Aran Davies, DevTeam.Space. (2018). *Pros and Cons of Hyperledger Fabric for Blockchain Networks*. Accessed: Aug. 12, 2018. [Online]. Available: https://www.devteam.space/blog/pros-and-cons-ofhyperledger-fabric-for-blockchain networks/

Ban, T., Q. (2019). Survey of Hyperledger Blockchain Frameworks.

Benet, J. (2014). "IPFS-Content Addressed, Versioned, P2P File System". arXiv:1407.3561v1.

Benhamouda, F., et al. (2019). "Supproting Private Data on Hyperledger Fabric with Secure Multiparty Computation". *IBM Journal of Research and Development, 63.*

Bonomi, S., Casini, M., & Ciccotelli, C. (2018). "B-CoC: A Blockchain-based Chain of Custody for Evidences Management in Digital Forensics". *arXive:1807.10359v1.*

BSN. (2014). *Teknologi Informasi - Teknik Keamanan-Pedoman Identifikasi, Pengumpulan, Akuisisi Dan Preservasi Bukti Digital (ISO/IEC 27037:2012)*. Jakarta: Badan Standardisasi Nasional.

Casey, E. et al. (2015). Leveraging CybOX™ to standardize representation and exchange of digital forensic information. *Proceedings of the Digital Forensic Research Conference, DFRWS 2015 EU, 12, (pp. 202-110).*

Giova, Giuliano. (2011). "Improving Chain of Custody in Forensic Investigation of Electronic Digital Systems." *International Journal of Computer Science and*

Network Security 11 (1): 1–9. http://paper.ijcsns.org/07_book/201101/20110101.pdf.

Granja, F., & Rafael, G. (2017). "The preservation of digital evidence and its admissibility in the court". *International Journal of Electronic Security and Digital Forensics, vol. 9, p1-18.*

Hackernoon. (2019). Hyperledger Fabric- The Most Popular Hyperledger Framework, https://hackernoon.com/hyperledger-fabric-the-most-popular-hyperledger-framework-b4485dea6a2c.

Hanafi, J. et al. (2021). "IPFSChain : Interplanetary File System and Hyperledger Fabric Collaboration for Chain of Custody and Digital Evidence Management". *International Journal of Computer Applications, vol. 183, (pp. 24-31).*

Hershensohn, J. (2005). "I.T. Forensics: the collection and presentation of digital evidence". Retrieved October 20, 2008, from http://icsa.cs.up.ac.za/issa/2005/Proceedings/Full/076_Article.pdf.

Hyperledger Fabric. (2019). Hyperledger Fabricdocs Documentation, Release master. https://readthedocs.org/projects/hyperledger-fabric/downloads/pdf/latest/

IBM. (2018). IBM Blockchain Platform. *Ibm.Com,* (April). Retrived from https://www.ibm.com/blockchain/platform.

Introduction – Hyperledger Composer, (2018). https://hyperledger.github.io/composer/latest/introduction/introduction.html. (Akses 02 Desember 2019).

Kao, D. et al. (2019). "The Governance of Digital Forensic Investigation in Law Enforcement Agencies". *International Conference on Advanced Communication Technology, ICACT, (pp. 61-65).*

Labs, Protocol. IPFS Is the Distributed Web. IPFS, https://ipfs.io/. Akses 07 Feb. 2021.

Leeds, G. S., & Marra, P. A. (2000). *Discovering and preserving electronic evidence: How to avoid spoliation pitfalls in the computer age.* Retrieved November 16, 2008, from http://www.spsk.com/Articles/artdscov.cfm.

Lone, A. H., & Mir, R. N. (2019). "Forensic-Chain: Blockchain based Digital Forensic Chain of Custody with PoC in Hyperledger Composer". *Elsevier, 28, (pp. 44-45).*

Meher, A. (2018). How IPFS Works. Retrived from Medium https://medium.com/@akshay_111meher/how-ipfs-works-545e1c890437. (Akses 23 Januari 2021)

Mougayar, W. (2016). *The Business Blockchain.* Wiley.

Nyaletey, E. (2019). “BlockIPFS - Blockchain-enabled interplanetary file system for forensic and trusted data traceability”. *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019, (pp. 18-25).*

Plessing, P. (2019). “Comparing Distributed Ledgers by Game of Thrones use case – Hyperledger (Fabric + Composer)”. https://medium.com/block42-blockchain-company/comparing-distributed-ledgers-by-game-of-thrones-use-case-hyperledger-fabric-composer-613782b7d529.

Prayudi, Y., Ashari, A., and Priyambodo, T. K. (2014). “Digital Evidence Cabinets: A Proposed Framework for Handling Digital Chain of Custody”. *Int. J. Comput. Appl., vol. 107, no. 9, pp. 30–36.*

Prayudi, Y., & SN, A. (2015). “Digital Chain of Custody: State of The Art”. *International Journal of Computer Applications, 114*(5), 1-9. https://doi.org/10.5120/19971-1856.

Putra, A., & Prayudi, Y. (2021). “Implementasi Multi Smart Contract pada Bukti Digital dan Chain of Custody dalam Meningkatkan Keamanan dan Integritas Bukti Digital”. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia), vol. 6, (pp. 98-108).*

Ratnasari, D., Prayudi, Y., & Sugiantoro, B. (2018). XML Approach for the Solution of Chain of Custody of Digital Evidence. International Journal of Computer Applications, 179(23), 20-25.

Rokkex. (2019). PoS, PoW, and 12 Other Blockchain Protocols You Didn’t Know About. https://medium.com/hackernoon/pos-pow-and-12-other-blockchain-protocols-you-didnt-know-about-3634b089d119?, (Akses 16 Desember 2019).

Santos, M., & Moura, E. (2019). Hands-On IoT Solutions with Blockchain. Brimingham-Mumbai: Packt Publishing.

Singhal, dkk., (2018). “Begining Blockchain : A Beginner’s Guide to Building Blockchain Solutions”. Appers. https://doi.org/10.1007/978-1-4842-3444-0.

SMITH & CROWN. (2019). Blockchain Technology: Comparing Ethereum, EOS & Hyperledger, Original Research. https://sci.smithandcrown.com/research/blockchain-technology-platforms#ethereum.

Syed, T., A., et al. (2019). “A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recomendations”. *IEEE Access, 7,* (pp. 176838-176869).

Thomas, D. S. (2004). *Legal methods of using computer forensics techniques for computer crime analysis and investigation*. Retrieved September 15, 2008, from http://www.iacis.org/iis/2004_iis/PDFfiles/ThomasForcht.pdf.

Wang, S., et al. (2018). "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems". *IEEE Access, 6,* (pp. 38437-38450).

Wibowo, D., F., H., Satria. 2019. "Perancangan dan Implementasi Teknologi *Blockchain* pada Sistem Pencatatan Hasil Rekapitulasi Pemilu Berdasarkan Formulir C1 Pindaian KPU". Tesis. Teknik Elektro, Institut Teknologi Bandung, Bandung.

Wüst, K., Gervais, A. (2017). "Do you need a Blockchain? IACR Cryptol". ePrint Archive 2017, 375.

Yunianto, Eko. 2019. "B-DEC: *Digital Evidence Cabinet* berbasis *Blockchain* untuk Manajemen Bukti". Tesis. Teknik Informatika, Universitas Islam Indonesia. Yogyakarta.

LAMPIRAN A

Source Code file **model.cto**

```
namespace org.example.empty

  //ASET
  asset CoC_Pencurian identified by CoCId{
  o String CoCId
  o String First_Responder
  o String File_Name
  o String Suspect
  o String Victim
  o String Date
  o String MD5
  o String SHA265
  o String CID_SHA2_256
  o String Location
  --> Participant issuer
  --> Participant owner
  }
  asset CoC_Penipuan identified by CoCId{
  o String CoCId
  o String First_Responder
  o String File_Name
  o String Suspect
  o String Victim
  o String Date
  o String MD5
  o String SHA265
  o String CID_SHA2_256
  o String Location
  --> Participant issuer
  --> Participant owner
  }
  asset CoC_Penghinaan identified by CoCId{
  o String CoCId
  o String First_Responder
  o String File_Name
  o String Suspect
  o String Victim
  o String Date
  o String MD5
  o String SHA265
  o String CID_SHA2_256
  o String Location
  --> Participant issuer
  --> Participant owner
  }
```

```
//PARTISIPAN
participant Officer identified by email {
o String email
o String firstName
o String lastName
}
participant First_Responder identified by email {
o String email
o String firstName
o String lastName
}
participant External identified by email {
o String email
o String firstName
o String lastName
o String Desc //Jaksa, Hakim, Investigator (dari orgnisasi yg lain) dll
}
participant Investigator identified by email {
o String email
o String firstName
o String lastName
}

//TRANSAKSI
transaction TransferCoC_Pencurian {
--> CoC_Pencurian coc
--> Participant issuer
--> Participant newOwner
}
transaction TransferCoC_Penipuan {
--> CoC_Penipuan coc
--> Participant issuer
--> Participant newOwner
}
transaction TransferCoC_Penghinaan {
--> CoC_Penghinaan coc
--> Participant issuer
--> Participant newOwner
}
```

Source Code file **permission.acl**

```
//OTORITAS SUPER ADMIN
rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system
resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
rule AdminOfficer { //Otoritas Admin terhadap partisipan Officer
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.Officer"
  action: ALLOW
}
rule AdminFirst_Responder { //Otoritas Admin terhadap partisipan FirstResponder
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.First_Responder"
  action: ALLOW
}
rule AdminInvestigator { //Otoritas Admin terhadap partisipan Investigator
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.Investigator"
  action: ALLOW
}
rule AdminExternal { //Otoritas Admin terhadap partisipan Extern
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.External"
  action: ALLOW
}
rule AdminCoC_Pencurian { //Otoritas Admin terhadap Dokument Chain oF Custody
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.CoC_Pencurian"
  action: ALLOW
}
rule AdminTransferCoC_Pencurian { //Otoritas Admin terhadap Dokument Chain oF
Custody
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.TransferCoC_Pencurian"
  action: ALLOW
}
rule AdminCoC_Penipuan { //Otoritas Admin terhadap Dokument Chain oF Custody
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.CoC_Penipuan"
  action: ALLOW
}
```

```
rule AdminTransferCoC_Penipuan { //Otoritas Admin terhadap Dokument Chain oF
Custody
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.TransferCoC_Penipuan"
  action: ALLOW
}
rule AdminCoC_Penghinaan { //Otoritas Admin terhadap Dokument Chain oF Custody
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.CoC_Penghinaan"
  action: ALLOW
}
rule AdminTransferCoC_Penghinaan { //Otoritas Admin terhadap Dokument Chain oF
Custody
  description: "Allow the auctioneer full access"
  participant: "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.example.empty.TransferCoC_Penghinaan"
  action: ALLOW
}

//-------------------------------------------------------+OTORITAS OFFICER (OR)
rule OfficerSystem { // OR terhadap system
  description: ""
  participant: "org.example.empty.Officer"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
rule OfficerFirst_Responder { // OR terhadap partisipan FR lainnya
  description: ""
  participant: "org.example.empty.Officer"
  operation: ALL
  resource: "org.example.empty.First_Responder"
  action: ALLOW
}
rule OfficerOfficer { // OR terhadap partisipan OR
  description: ""
  participant: "org.example.empty.Officer"
  operation: READ
  resource: "org.example.empty.Officer"
  action: ALLOW
}
rule OfficerInvestigator { // OR terhadap partisipan IR
  description: ""
  participant: "org.example.empty.Officer"
  operation: ALL
  resource: "org.example.empty.Investigator"
  action: ALLOW
}
rule OfficerExternal { // OR terhadap partisipan ER
  description: ""
  participant: "org.example.empty.Officer"
  operation: ALL
  resource: "org.example.empty.External"
  action: ALLOW
}
```

```
rule OfficerCoC_Penipuan { // OR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.Officer"
  operation: READ, DELETE
  resource: "org.example.empty.CoC_Penipuan"
  action: ALLOW
}
rule OfficerCoC_Penghinaan { // OR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.Officer"
  operation: READ, DELETE
  resource: "org.example.empty.CoC_Penghinaan"
  action: ALLOW
}
rule OfficerCoC_Pencurian { // OR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.Officer"
  operation: READ, DELETE
  resource: "org.example.empty.CoC_Pencurian"
  action: ALLOW
}
rule OfficerTransferCoC_Penipuan { // OR terhadap transaksi moveEvidence
  description: ""
  participant: "org.example.empty.Officer"
  operation: READ
  resource: "org.example.empty.TransferCoC_Penipuan"
  action: ALLOW
}



//-----------------------------------------------+OTORITAS FIRST RESPONDENT (FR)
rule First_ResponderSystem { // FR terhadap system
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
rule First_ResponderFirst_Responder { // FR terhadap partisipan FR lainnya
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: READ
  resource: "org.example.empty.First_Responder"
  action: ALLOW
}
rule First_ResponderOfficer { // FR terhadap partisipan OR
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: READ
  resource: "org.example.empty.Officer"
  action: ALLOW
}
```

```
rule First_ResponderInvestigator { // FR terhadap partisipan IR
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: READ
  resource: "org.example.empty.Investigator"
  action: ALLOW
}
rule First_ResponderExternal { // FR terhadap partisipan ER
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: READ
  resource: "org.example.empty.External"
  action: ALLOW
}
rule First_ResponderCoC_Penipuan { // FR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: CREATE, READ
  resource: "org.example.empty.CoC_Penipuan"
  action: ALLOW
}
rule First_ResponderCoC_Penghinaan { // FR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: CREATE, READ
  resource: "org.example.empty.CoC_Penghinaan"
  action: ALLOW
}
rule First_ResponderCoC_Pencurian { // FR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: CREATE, READ
  resource: "org.example.empty.CoC_Pencurian"
  action: ALLOW
}
rule First_ResponderTransferCoC_Penipuan { // FR terhadap transaksi moveEvidence
  description: ""
  participant: "org.example.empty.First_Responder"
  operation: READ
  resource: "org.example.empty.TransferCoC_Penipuan"
  action: ALLOW
}

//-------------------------------------------------+OTORITAS INVESTIGATOR (FR)
rule InvestigatorSystem { // IR terhadap system
  description: ""
  participant: "org.example.empty.Investigator"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
rule InvestigatorFirst_Responder { // IR terhadap partisipan FR lainnya
  description: ""
  participant: "org.example.empty.Investigator"
  operation: READ
  resource: "org.example.empty.First_Responder"
  action: ALLOW
}
rule InvestigatorOfficer { // IR terhadap partisipan OR
  description: ""
  participant: "org.example.empty.Investigator"
  operation: READ
```

```
  resource: "org.example.empty.Officer"
  action: ALLOW
}
rule InvestigatorInvestigator { // IR terhadap partisipan IR
  description: ""
  participant: "org.example.empty.Investigator"
  operation: READ
  resource: "org.example.empty.Investigator"
  action: ALLOW
}
rule InvestigatorExternal { // IR terhadap partisipan ER
  description: ""
  participant: "org.example.empty.Investigator"
  operation: READ
  resource: "org.example.empty.External"
  action: ALLOW
}
rule InvestigatorCoC_Penipuan { // IR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.Investigator"
  operation: READ, UPDATE
  resource: "org.example.empty.CoC_Penipuan"
  action: ALLOW
}
rule InvestigatorCoC_Penghinaan { // IR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.Investigator"
  operation: READ, UPDATE
  resource: "org.example.empty.CoC_Penghinaan"
  action: ALLOW
}
rule InvestigatorCoC_Pencurian { // IR terhadap DOKUMEN CoC
  description: ""
  participant: "org.example.empty.Investigator"
  operation: READ, UPDATE
  resource: "org.example.empty.CoC_Pencurian"
  action: ALLOW
}
//---------------TRANSAKSI
rule InvestigatorTransferCoC_Penipuan { // IR terhadap transaksi moveEvidence
  description: ""
  participant: "org.example.empty.Investigator"
  operation: CREATE, READ
  resource: "org.example.empty.TransferCoC_Penipuan"
  action: ALLOW
}
rule InvestigatorTransferCoC_Penghinaan { // IR terhadap transaksi moveEvidence
  description: ""
  participant: "org.example.empty.Investigator"
  operation: CREATE, READ
  resource: "org.example.empty.TransferCoC_Penghinaan"
  action: ALLOW
}
rule InvestigatorTransferCoC_Pencurian { // IR terhadap transaksi moveEvidence
  description: ""
  participant: "org.example.empty.Investigator"
  operation: CREATE, READ
  resource: "org.example.empty.TransferCoC_Pencurian"
  action: ALLOW
}
```

```
rule InvestigatorTransferCoC_Pencurian { // FR terhadap transaksi moveEvidence
  description: ""
  participant: "org.example.empty.Investigator"
  operation: CREATE, READ
  resource: "org.example.empty.TransferCoC_Pencurian"
  action: ALLOW
}
```

Source Code file **logic.js**

```
/**
 * Trade a asset to a new player
 * @param {org.example.empty.TransferCoC_Pencurian} moveAset - the trade asset
transaction
 * @transaction
 */
 async function CoC_Pencurian(CoC_Pencurian) { // eslint-disable-line no-unused-
vars
    CoC_Pencurian.coc.issuer = CoC_Pencurian.coc.owner;
    CoC_Pencurian.coc.owner = CoC_Pencurian.newOwner;
    const assetRegistry = await
getAssetRegistry('org.example.empty.CoC_Pencurian');
    await assetRegistry.update(CoC_Pencurian.coc);
   }

   /**
    * Trade a asset to a new player
    * @param {org.example.empty.TransferCoC_Penipuan} moveAset - the trade asset
transaction
    * @transaction
    */
   async function CoC_Penipuan(CoC_Penipuan) { // eslint-disable-line no-unused-
vars
    CoC_Penipuan.coc.issuer = CoC_Penipuan.coc.owner;
    CoC_Penipuan.coc.owner = CoC_Penipuan.newOwner;
    const assetRegistry = await
getAssetRegistry('org.example.empty.CoC_Penipuan');
    await assetRegistry.update(CoC_Penipuan.coc);
   }

   /**
    * Trade a asset to a new player
    * @param {org.example.empty.TransferCoC_Penghinaan} moveAset - the trade
asset transaction
    * @transaction
    */
   async function CoC_Penghinaan(CoC_Penghinaan) { // eslint-disable-line no-
unused-vars
    CoC_Penghinaan.coc.issuer = CoC_Penghinaan.coc.owner;
    CoC_Penghinaan.coc.owner = CoC_Penghinaan.newOwner;
    const assetRegistry = await
getAssetRegistry('org.example.empty.CoC_Penghinaan');
    await assetRegistry.update(CoC_Penghinaan.coc);
   }
```